

UNIVERSIDAD CARLOS III

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA INFORMÁTICA



TRABAJO DE FIN DE GRADO

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE
SENSORES EN TELÉFONOS INTELIGENTES

Autor: José Luis Jiménez Fontenla

Tutores: Jorge Blasco Alís
José María de Fuentes García-Romero de Tejada

Marzo 2014

AGRADECIMIENTOS

Con este Trabajo de Fin de Grado termina otra etapa en mi vida, etapa que cuando aún me encontraba en el colegio estaba deseando que llegara, y nada más comenzar, estaba deseando que terminara.

Dejo atrás muchos compañeros y grandes profesores, pero a la nueva etapa que comienzo me llevo conmigo muy buenos recuerdos y anécdotas. Gracias a amigos como Berni, el paso por la Universidad ha sido más llevadero, sobre todo porque son casi 21 años desde que nos conocemos. Hasta ahora todas las etapas las hemos pasado juntos; y en esta última, hemos compartido muchas horas de prácticas interminables y noches sin dormir. ¿Coincidiremos en la siguiente?

Como no mencionar a Pilar, todo el estrés y agobio que ha tenido que aguantar por mi parte durante todos estos años y aún así, ahí sigue a mi lado. Tú sí que vas a estar seguro en el resto de etapas de mi vida. Nuestras etapas.

Gracias también a mi familia y amigos, los primeros por aguantar mi desastre en mi “cueva” (la habitación donde estudio) y mis frases durante todo el día de cansancio. A mis amigos por seguir considerándome uno de ellos a pesar de que ahora soy un abuelo como ellos dicen.

Por último y no menos importante, gracias a mis tutores Jorge y Chema; Chema y Jorge por lo que me han estado ayudando y guiando todos estos meses.

Ahora sí, comienza nuestra etapa.

RESUMEN

Con el tiempo, el uso de los móviles ha sufrido grandes cambios sobre todo de funcionalidad y uso; dispositivos que en un principio estaban pensados para realizar llamadas o enviar mensajes de texto, hoy en día esa solo es la mínima parte de las características que pueden tener los llamados Smartphones o teléfonos inteligentes actuales. Inteligentes ya que, gracias a la tecnología que incorporan como por ejemplo los sensores, pueden reaccionar ante cambios que se producen en nuestro entorno sin tener que interactuar con ellos. En estos últimos tiempos se está avanzando más en este aspecto, fabricándose complementos para los Smartphones que hacen uso de estos sensores, por ejemplo para registrar nuestra actividad vital.

Este Trabajo de Fin de Grado permite, a través de una aplicación para dispositivos móviles con sistema operativo Android, recoger el valor de diferentes sensores y GPS y enviarlos periódicamente a un servidor web que estará gestionado por un administrador. Para poder almacenar los datos y gestionarlos también se desarrolla una aplicación web alojada en Google App Engine.

Para que el usuario del dispositivo móvil disponga de total libertad a la hora de gestionar la aplicación, se le da la posibilidad de parar cuando desee su ejecución. Gracias a la aplicación web, el administrador de la misma puede en cualquier momento modificar diferentes parámetros de las aplicaciones instaladas en los dispositivos, como son el intervalo de envío de datos al servidor o el intervalo de captura de los datos; además puede consultar toda la información enviada al servidor de cualquier usuario registrado en la aplicación.

Palabras clave: sensores, Smartphone, captura de datos.

ABSTRACT

In the course of time, the use of mobile phones has undergone big changes especially in functionality and use. These devices were originally designed to make calls or send text messages. Nowadays that is only a small part of the functionality that the new type of mobiles phones called Smartphones have. I mean smart ones as with the great progress in technology such as incorporating sensors, they can react to changes that happen in our environment without having to interact with them. In these lately days, technology is advancing at a dizzy speed, as manufacturing smartphone's accessories which make use of these sensors, for example to record our vital activity.

This Final Project Degree allow us to collect the value of different sensors and GPS and periodically send them to a web application which will be managed by administrator through an application for mobile devices with Android operating system. In order to store and manage data, a hosted web application it is also developed in Google App Engine.

Device's users have the possibility to stop execution whenever they want, I mean they have full freedom to manage the application. Thanks to the web application, the administrator may at any time modify various parameters of the applications installed on devices such as the interval for sending data to the server or the range of data capture. Moreover they are also able to view all data sent to the server of any user registered in the application.

Keywords: sensors, Smartphone, data capture.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	3
RESUMEN	4
ABSTRACT	5
ÍNDICE DE FIGURAS.....	9
ÍNDICE DE TABLAS	11
ACRÓNIMOS Y DEFINICIONES	14
1. INTRODUCCIÓN	15
1.1. MOTIVACIÓN	15
1.2. OBJETIVO.....	16
1.3. ASPECTOS LEGALES	17
1.4. ESTRUCTURA DEL DOCUMENTO	18
2. ESTADO DEL ARTE.....	19
2.4. SENSORES	19
2.4.1. ACELERÓMETRO	21
2.4.2. POSICIÓN	23
2.4.3. GIROSCOPIO	24
2.4.4. GPS (<i>Global Positioning System</i>).....	25
2.5. APLICACIONES ACTUALES	26
2.5.1. MOVES.....	26
2.5.2. CAMISETAS INTELIGENTES.....	26
2.5.3. JUEGOS.....	27
2.5.4. EFICIENCIA DE RECURSOS EN DISPOSITIVOS.....	27
3. ANÁLISIS	28
3.4. CASOS DE USO	28
3.5. REQUISITOS DE SOFTWARE.....	33
3.5.1. REQUISITOS FUNCIONALES.....	33
3.5.2. REQUISITOS NO FUNCIONALES	38
3.6. TRAZABILIDAD DE REQUISITOS DE SOFTWARE Y CASOS DE USO	39
3.7. ENTORNO DE DESARROLLO	40
3.7.1. JAVA	40
3.7.2. BOOTSTRAP	40
3.7.3. ECLIPSE IDE	40
3.8. TECNOLOGÍA	41
3.8.1. GOOGLE APP ENGINE.....	41
3.8.2. GOOGLE CLOUD ENDPOINTS.....	42
3.8.3. PLATAFORMA.....	43
3.5.4. OTRAS TECNOLOGÍAS	44
3.9. DIAGRAMA DE ESTADOS	45
3.10. CATÁLOGO DE PRUEBAS	48
4. DISEÑO	56
4.4. ARQUITECTURA	56
4.5. PATRÓN DE DISEÑO	57
4.5.1. APLICACIÓN WEB.....	58

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN TELÉFONOS INTELIGENTES

4.5.2. APLICACIÓN MÓVIL.....	59
4.6. DIAGRAMAS DE ACTIVIDAD.....	60
4.6.1. APLICACIÓN WEB.....	61
4.6.2. APLICACIÓN MOVIL.....	65
4.7. DIAGRAMA DE COMPONENTES	69
4.8. DIAGRAMA DE CLASES	72
4.9. INTERFACES.....	74
4.9.1. APLICACIÓN MÓVIL.....	74
4.9.2. APLICACIÓN WEB.....	76
5. IMPLEMENTACIÓN Y PRUEBAS	79
5.4. CONCURRENCIA.....	79
5.5. COMUNICACIÓN CON APLICACIÓN MÓVIL.....	80
5.6. ALMACENAMIENTO DE DATOS	84
5.7. PRIVACIDAD DEL USUARIO	86
5.8. REGISTRO Y LECTURA DE SENSORES	88
5.9. RESULTADOS DE LAS PRUEBAS	91
6. CONCLUSIONES Y TRABAJOS FUTUROS.....	92
6.4. CONCLUSIONES PERSONALES	93
6.5. TRABAJOS FUTUROS.....	93
REFERENCIAS	95
ANEXO A – GESTIÓN DE PROYECTO.....	97
PLANIFICACIÓN REAL	101
ANEXO B – MANUAL DE USUARIO.....	104
A.1 APLICACIÓN WEB	104
A.2 APLICACIÓN MÓVIL	108

ÍNDICE DE FIGURAS

Imagen 1. Sistema de coordenadas de sensores aceleración y giroscopio (Android developer).	20
Imagen 2. Sistema de coordenadas sensor de posición (3)	21
Imagen 3. Diagrama de casos de uso.	28
Imagen 4. Arquitectura de <i>Google Cloud Endpoints</i> (14).	42
Imagen 5. Comparación de versiones iOS vs Android (15).	43
Imagen 6. Diagrama de estados aplicación móvil.	46
Imagen 7. Zoom de los modos contenidos en el estado Específico.	46
Imagen 8. Arquitectura de sistemas.	56
Imagen 9. Arquitectura aplicación web (17).	58
Imagen 10. Arquitectura aplicación Android (18).	59
Imagen 11. Diagrama de actividad de autenticación en el sistema.	61
Imagen 12. Diagrama de actividad para modificar parámetros.	62
Imagen 13. Diagrama de actividad para la consulta de datos.	63
Imagen 14. Diagrama de actividad cierre de sesión.	64
Imagen 15. Diagrama de actividad para registrarse en el sistema.	65
Imagen 16. Diagrama de actividad para darse de baja.	66
Imagen 17. Diagrama de actividad para cambiar modo de ejecución.	67
Imagen 18. Diagrama de actividad para parar la lectura de sensores.	68
Imagen 19. Diagrama de componentes del sistema.	69
Imagen 20. Diagrama de componentes aplicación móvil.	70
Imagen 21. Diagrama de componentes aplicación web.	71
Imagen 22. Diagrama de clases aplicación móvil.	72
Imagen 23. Diagrama de clases aplicación web.	73
Imagen 24. Pantalla principal.	74
Imagen 25. Pantalla para volver a registrarse.	75
Imagen 26. Pantalla de acceso a la aplicación web.	76
Imagen 27. Pantalla administración.	77
Imagen 28. Pantalla para consultar datos.	77
Imagen 29. Resultados consulta.	78
Imagen 30. Pantalla modificación de clientes.	78
Imagen 31. Parar lectura de sensores, GPS y dormirse.	80
Imagen 32. Ejemplo de identificador de Google Cloud Messaging	81
Imagen 33. Formulario para modificar parámetros de clientes.	81
Imagen 34. Envío de datos usando Google Cloud Messaging I.	82
Imagen 35. Envío de datos usando Google Cloud Messaging II.	83
Imagen 36. Código endpoint usuario.	85
Imagen 37. Código para la generación de números.	87
Imagen 38. Código para modificar latitud y longitud.	88
Imagen 39. Código del registro de un sensor.	89
Imagen 40. Código para leer los valores de un sensor.	90
Imagen 41. Diagrama de Gantt - Planificación.	97
Imagen 42. Diagrama de Gantt - Real.	101
Imagen 43. Pantalla de acceso a la aplicación web.	104

Imagen 44. Pantalla administración.....	105
Imagen 45. Acceso incorrecto.....	105
Imagen 46. Pantalla para consultar datos.....	106
Imagen 47. Resultados consulta.....	106
Imagen 48. Pantalla modificación de clientes.....	107
Imagen 49. Pantalla principal.....	108
Imagen 50. Baja de aplicación móvil.....	109
Imagen 51. Pantalla de registro.....	110

ÍNDICE DE TABLAS

Tabla 1 . Aplicaciones de los sensores.	20
Tabla 3. Caso de uso 01.	29
Tabla 4. Caso de uso 02	30
Tabla 5. Caso de uso 03.	30
Tabla 6. Caso de uso 05.	31
Tabla 7. Caso de uso 06	31
Tabla 8. Caso de uso 07.	32
Tabla 9. Caso de uso 08.	32
Tabla 10. Requisito de Software Funcional 01.....	33
Tabla 11. Requisito de Software Funcional 02.....	34
Tabla 12. Requisito de Software Funcional 03.....	34
Tabla 13. Requisito de Software Funcional 04.....	34
Tabla 14. Requisito de Software Funcional 05.....	34
Tabla 15. Requisito de Software Funcional 06.....	34
Tabla 16. Requisito de Software Funcional 07.....	35
Tabla 17. Requisito de Software Funcional 08.....	35
Tabla 18. Requisito de Software Funcional 09.....	35
Tabla 19. Requisito de Software Funcional 10.....	35
Tabla 20. Requisito de Software Funcional 11.....	35
Tabla 21. Requisito de Software Funcional 12.....	36
Tabla 22. Requisito de Software Funcional 13.....	36
Tabla 23. Requisito de Software Funcional 14.....	36
Tabla 24. Requisito de Software Funcional 15.....	36
Tabla 25. Requisito de Software Funcional 16.....	36
Tabla 26. Requisito de Software Funcional 17.....	37
Tabla 27. Requisito de Software Funcional 18.....	37
Tabla 28. Requisito de Software Funcional 19.....	37
Tabla 29. Requisito de Software Funcional 20.....	37
Tabla 30. Requisito de Software Funcional 01.....	38
Tabla 31. Requisito de Software Funcional 02.....	38
Tabla 32. Requisito de Software no Funcional 03.	38
Tabla 33. Requisito de Software no Funcional 04.	38
Tabla 34. Requisito de Software no Funcional 05.	38
Tabla 35. Requisito de Software no Funcional 06.	39
Tabla 36. Trazabilidad requisitos de software y casos de uso.	39
Tabla 37. Top 4 de sistemas operativos 3º trimestre 2013 (unidades en millones) (16). ..	44
Tabla 38. Transiciones y estados de la aplicación móvil.	47
Tabla 39. Checklist de resultados de pruebas.	49
Tabla 40. Alta automática en el sistema.....	49
Tabla 41. Baja en el sistema.....	50
Tabla 42. Borrado de datos del sistema.	50
Tabla 43. Iniciar lectura de sensores.....	50
Tabla 44. Cambiar modo de ejecución.....	51
Tabla 45. Cambio de modo no permitido.	51

Tabla 46. Detener lectura de sensores.....	51
Tabla 47. Pedir activación de GPS.....	51
Tabla 48. Acceso a la aplicación web satisfactoria.	52
Tabla 49. Acceso a la aplicación web erróneo	52
Tabla 50. Cierre de sesión del administrador.....	52
Tabla 51. Consulta de datos de un usuario en la aplicación web.	53
Tabla 52. Modificación de parámetros.	53
Tabla 53. Descarga de datos.....	54
Tabla 54. Error en valor de parámetro a modificar I.	54
Tabla 55. Error en valor de parámetro a modificar II.....	55
Tabla 56. Error de consulta de datos.....	55
Tabla 57. Comunicación aplicaciones.	55
Tabla 58. Elementos diagrama de actividad.	60
Tabla 59. Resultado de las pruebas ejecutadas.....	91
Tabla 60. Perfiles de personal.	98
Tabla 61. Coste de personal - Planificado.....	98
Tabla 62. Coste de equipos - Planificado.	99
Tabla 63. Coste de software - Planificado.....	99
Tabla 64. Costes generales - Planificado.....	100
Tabla 65. Resumen costes - Planificado.	100
Tabla 66. Coste de personal - Real.....	101
Tabla 67. Coste de equipos - Real.....	102
Tabla 68. Coste de software - Real.....	102
Tabla 69. Costes generales - Real	102
Tabla 70. Diferencia de costes.....	103

ACRÓNIMOS Y DEFINICIONES

- ⇒ API: Application Programming Interface. Librerías que aportan diferentes funcionalidades para usar en los programas
- ⇒ Framework: Librería que facilita el desarrollo de la programación aportando diferentes recursos.
- ⇒ GLONASS: Sistema Global de Navegación por Satélite (GNSS).
- ⇒ GO: Lenguaje de programación basado en C desarrollado por Google.
- ⇒ GPS: Sistema de Posicionamiento Global.
- ⇒ Hardware: Se refiere a las partes físicas de un dispositivo.
- ⇒ Http: Hypertext Transfer Protocol. Protocolo para acceder a páginas web.
- ⇒ iOS: Sistema operativo desarrollado por Apple presente en los dispositivos iPhone.
- ⇒ PHP: PHP Hypertext Pre-processor. Se trata de un lenguaje de programación del lado servidor.
- ⇒ Python: Lenguaje de programación
- ⇒ SDK: Software Development Kit. Es un paquete de ayuda para desarrollar programas.
- ⇒ Sistema operativo: Se trata del software base de un dispositivo que se encarga de gestionar los recursos hardware.
- ⇒ SMS: *Short Message Service*. Servicio de mensajes cortos.
- ⇒ Software: Hace referencia a los módulos no físicos de un dispositivo como los programas.
- ⇒ SQLite: Base de datos relacional usada en dispositivos móviles
- ⇒ SSL: Secure Sockets Layer. Se trata de un protocolo para intercambiar datos cifrados a través de Http.
- ⇒ UML: Unified Modeling Language. Lenguaje usado para modelar sistemas.
- ⇒ WiFi: Tecnología inalámbrica para acceder a internet.
- ⇒ XML: eXtensible Markup Language. Lenguaje que permite especificar la estructura y significado de un documento usando etiquetas.

1. INTRODUCCIÓN

En este primer capítulo se realiza una pequeña introducción a la temática del Trabajo de Fin de Grado, comentando qué es lo que ha motivado al desarrollo del mismo y los objetivos que se han marcado para llegar a la meta marcada.

1.1. MOTIVACIÓN

Con la aparición de los Smartphones, el uso de los dispositivos móviles se ha convertido en una necesidad para muchas de las acciones que llevamos a cabo en nuestro día a día y una fuente de información muy potente en base a la cual tomamos muchas decisiones.

La tendencia es que nuestros dispositivos tomen esas decisiones por nosotros aprendiendo de nuestro comportamiento o anticipándose a nuestras necesidades. Esto se consigue gracias a la creciente integración de diferentes sensores en esos aparatos. Dispositivos que en un principio estaban diseñados para realizar llamadas, hoy en día, con su evolución, son capaces de detectar, por ejemplo, si estamos parados o en movimiento y de ese modo actuar de una forma u otra, o tener habilitadas ciertas funciones y otras no.

Estudiando esta evolución, los sensores tienen aún mucho que decir y aportar en el mundo de los dispositivos móviles que nos acompañan a diario, ya que se pueden usar en un sinfín de aplicaciones que hoy en día nos parece hasta difícil de imaginar. Pero... ¿quién nos iba a decir que, a día de hoy, un dispositivo detectaría cuando hemos descansado lo suficiente para decidir despertarnos?

1.2. OBJETIVO

El objetivo principal de este proyecto consiste en desarrollar una aplicación móvil para el sistema operativo Android que se encargue de recoger los valores leídos por diferentes sensores del dispositivo y, a continuación, enviarlos de forma periódica a un servidor web que se encargará de almacenarlos, así que, complementariamente, también se tiene como objetivo el desarrollo de un servidor web para el almacenamiento de datos y la gestión de los diferentes dispositivos móviles con la aplicación instalada.

Para cumplir estos objetivos se deberá:

- ✓ Diseñar y desarrollar una aplicación Android en lenguaje Java capaz de obtener datos de los sensores. Será necesario:
 - Registrar al usuario en la aplicación.
 - Obtener los sensores disponibles en el dispositivo móvil.
 - Obtener los datos de los sensores más influyentes.
 - Enviar los datos a un servidor web.
- ✓ Diseñar y desarrollar una aplicación web que sea capaz de comunicarse con la aplicación móvil y almacenar datos con los siguientes subobjetivos:
 - Recibir peticiones de registro desde la aplicación móvil que puede estar instalada en varios dispositivos.
 - Almacenar datos recibidos desde la aplicación móvil.
 - Consultar los datos almacenados en la base de datos.
 - Modificar parámetros de las aplicaciones de los dispositivos móviles.

1.3. ASPECTOS LEGALES

En un sistema como el que se pretende desarrollar, hay que tener en cuenta diferentes aspectos para estar siempre dentro de la legalidad. Sobre todo, se debe tener presente la Ley Orgánica de Protección de Datos, para proteger la información personal de cada uno de los usuarios del sistema.

Los datos de este carácter están definidos legalmente como “Cualquier información numérica, alfabética, fotográfica, acústica o de cualquier otro tipo concerniente a personas físicas identificadas o identificables.” (1). Se considera que una persona es identificada cuando se sabe a quién pertenece ese dato y se le considera identificable cuando no se sabe a quién pertenece, pero se podría averiguar sin demasiados esfuerzos.

Por ello, en las aplicaciones a desarrollar, tenemos que tener en cuenta la Ley Orgánica de Protección de Datos y no utilizar ningún dato de carácter personal como son el número de teléfono, DNI, IMEI (*International Mobile Equipment Identity*, Identidad Internacional de Equipo Móvil)) o usuario y contraseña sin almacenarlos cifrados, de forma que no se pudiera saber el contenido real. Para evitar problemas, la aplicación móvil realiza el registro en el servidor web generando un número aleatorio de forma que el usuario nunca podría ser identificado ni identificable.

1.4. ESTRUCTURA DEL DOCUMENTO

El presente documento cuenta con una serie de apartados clasificados según su temática:

1. Introducción: Se realiza una aproximación de la finalidad del proyecto con las motivaciones que llevan a realizarlo, así como la presentación de su estructura.
2. Estado del arte: Se realiza una investigación y análisis sobre los sensores y de las actuales tendencias en la utilización de los mismos.
3. Análisis: Se detallan los requisitos de software identificados que debe cumplir la aplicación móvil y web, así como los diferentes casos de usos que se contemplan en ambos sistemas. Por último se analiza la relación entre ambos, y el catálogo de pruebas a realizar; además de realizar un estudio de las tecnologías y entornos de desarrollo utilizados en la realización del proyecto.
4. Diseño: Se presentan las decisiones tomadas para desarrollar la aplicación móvil y web con diferentes diagramas UML.
5. Implementación y pruebas: Se detallan los aspectos más importantes en el desarrollo de ambos sistemas así como los principales problemas encontrados y las soluciones implantadas. Por último se muestran los resultados obtenidos en las pruebas realizadas.
6. Conclusiones y trabajos futuros: Se resume el trabajo realizado, y las sensaciones experimentadas durante su desarrollo aportando ideas para su continuación.
7. Anexo A – Gestión de proyecto: Se presenta tanto la planificación establecida para el desarrollo del proyecto como el coste del mismo.
8. Anexo B – Manual de usuario: Se explican las diferentes pantallas que componen las aplicaciones desarrolladas y cómo interactuar con cada una de ellas.

2. ESTADO DEL ARTE

En esta sección se presenta la base teórica en la que se sustenta el Trabajo de Fin de Grado ofreciendo una visión global acerca de los temas que se van a tratar. En primer lugar, se realizará una introducción a los sensores, continuando con la descripción y funcionamiento de los principales que se encuentran en los dispositivos móviles. Por último se comentarán aplicaciones actuales de los sensores en diferentes campos.

2.4. SENSORES

Los sensores son pequeños dispositivos que se encargan de recoger información del medio en el que se encuentren y transformar dicha información en señales eléctricas que posteriormente serán interpretadas y tratadas por otros dispositivos. Estas señales son generadas a partir de las propiedades de los materiales de los que este hecho el sensor; ya que, dependiendo del material del que esté fabricado, éste va a sufrir una serie de cambios por las propiedades del medio (aumento de temperatura, presión...).

Según el parámetro a medir, se pueden clasificar los sensores de la siguiente manera: sensores de desplazamiento, de presión, de temperatura, de velocidad y de aceleración.

Debido a su bajo coste y simplicidad, los sensores están muy presentes en muchos de los procesos industriales de cualquier empresa, ya que siempre hay algún proceso en el que es necesario medir, decidir y actuar según los resultados obtenidos.

Ámbito	Aplicación
Militar	Visión nocturna, sonar.
Médico	Dispositivos terapéuticos implantables.
Automóvil	Asistencia de navegación, detectar deslizamiento.
Alimentario	Temperatura de cámaras frigoríficas.

Tabla 1 . Aplicaciones de los sensores.

A continuación se realizará una aproximación más detallada de los sensores que tienen, hoy en día, más relevancia en los dispositivos móviles, así como su funcionamiento. Antes de comenzar, es interesante conocer el sistema de coordenadas de cada uno de los sistemas.

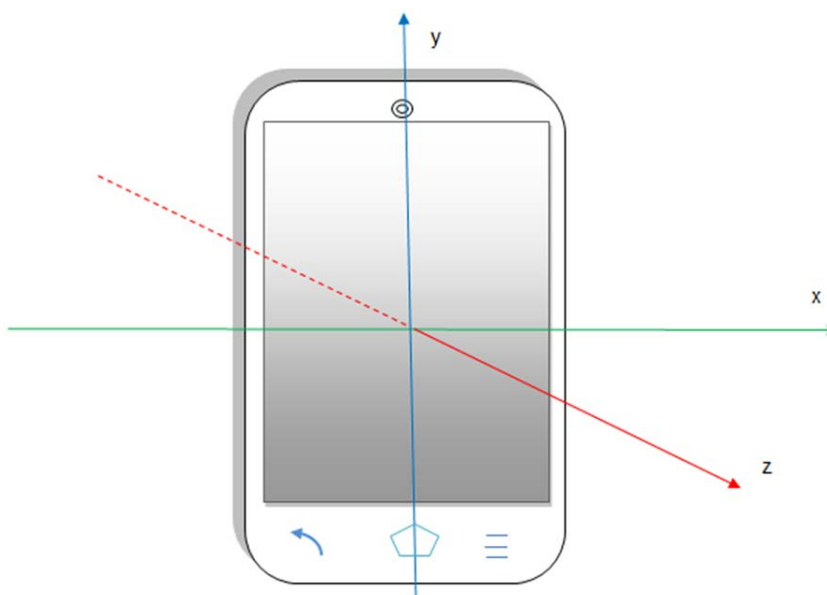


Imagen 1. Sistema de coordenadas de sensores aceleración y giroscopio (Android developer).

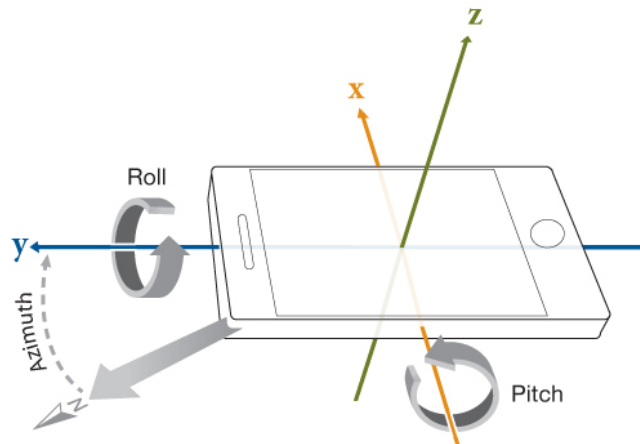


Imagen 2. Sistema de coordenadas sensor de posición (3) .

En la imagen anterior hay que tener en cuenta tres conceptos importantes: el *azimut*, *pitch* y *roll*.

- *Azimut*: Es el ángulo entre el eje “Y” y el norte magnético. El rango está entre 0 y 360 grados.
- *Pitch*: Se trata de la inclinación que tiene el dispositivo girando sobre el eje “X” respecto al eje “Z”.
- *Roll*: Es ángulo existente entre el móvil y el eje “X” cuando gira sobre el eje “Y”.

2.4.1. ACELERÓMETRO

Los acelerómetros son pequeños dispositivos usados para medir la aceleración además del ángulo de inclinación, rotación, vibración, choque y gravedad de un dispositivo (1). Existen varios tipos de sensores de aceleración dependiendo de la aplicación y uso que se les vaya a dar.

Su funcionamiento está basado en la Segunda Ley de Newton de 1687 en la que se describe el comportamiento de una masa cuando sufre una aceleración, obteniendo la fuerza que está experimentando $F = m \cdot a$, donde “F” es la fuerza, “a” es la aceleración y “m” es la masa (2).

Los acelerómetros han pasado de estar dedicados a un uso industrial a estar presentes en dispositivos tales como Smartphones, portátiles, etc. Nos centraremos principalmente en su uso en teléfonos móviles.

En un dispositivo móvil, el acelerómetro nos permite detectar el movimiento y el giro, es decir, detectar su posición y los movimientos que se realizan sobre el mismo. Su uso en Smartphones ofrece un amplio abanico de aplicaciones, actualmente se está innovando en la tecnología de estos dispositivos por medio del uso del acelerómetro. El software del dispositivo debe traducir los datos proporcionados por el sensor.

Cuando se aplica a un teléfono móvil, un acelerómetro puede identificar en qué posición se encuentra el teléfono y, en base a ello, rotar la pantalla de forma automática según la orientación del dispositivo.

Otra de las principales aplicaciones en los móviles, es utilizar el acelerómetro junto con la cámara, de forma que siempre sea posible posicionar una fotografía. El teléfono móvil incluye un acelerómetro con el que es posible detectar si la fotografía es tomada en horizontal y en vertical, grabando ese dato en la imagen y permitiendo verla con su orientación correcta (3).

Uno de los usos más extendidos de estos sensores en los dispositivos móviles es para los juegos, en los que el usuario no necesita interactuar con botones colocados en la pantalla sino que basta con realizar movimientos en el dispositivo produciendo diferentes acciones en el juego.

Además, gran cantidad de aplicaciones móviles hacen uso del acelerómetro. Por ejemplo, una de ellas consiste en evitar que el teléfono móvil sea utilizado sin consentimiento del usuario, de forma que hace sonar una alarma cada vez que el teléfono detecta un cierto movimiento (7). El uso del acelerómetro es fundamental en este tipo de aplicaciones, éste detectará una aceleración distinta de cero al pasar el teléfono móvil del reposo al movimiento. Gracias al uso de este sensor, otras

aplicaciones permiten leer SMS o incluso controlar la reproducción de la música y cambiar canciones simplemente agitando el teléfono (8).

2.4.2. POSICIÓN

Un sensor de posición es aquel que permite determinar la ubicación de un dispositivo móvil. Podemos diferenciar entre sensores de diferentes características que realizan dicha función; por un lado encontramos los sensores de campo magnético terrestre, y por otro, los sensores de orientación. Existe un tercer sensor conocido como el sensor de proximidad que proporciona la distancia entre el dispositivo y otro objeto.

El sensor de campo magnético terrestre y el sensor de proximidad están basados en hardware. La mayoría de los teléfonos móviles incluyen un sensor de campo magnético terrestre, que mide su intensidad del campo magnético de la tierra a lo largo de los ejes x, y, z. Su uso más importante en Smartphones consiste en la implementación de la brújula. Asimismo, la mayoría de los teléfonos móviles suelen llevar incorporado un sensor de proximidad, cuya función es detectar objetos a corta distancia. Dicho dispositivo principalmente se utiliza para conocer si la persona sostiene el teléfono móvil cerca de la cara, de forma que podrá determinar si el usuario mantiene una llamada telefónica y se procederá al bloqueo de la pantalla táctil durante la misma. Sin embargo, el sensor de orientación está basado en software y obtiene sus datos del acelerómetro y del sensor de campo magnético terrestre.

La detección de ubicaciones geográficas precisas de un determinado dispositivo se obtiene a través de los sensores de posición. Gracias a la combinación de un sensor de campo magnético terrestre con un acelerómetro se puede determinar la posición relativa de un cierto dispositivo con respecto a un polo magnético. También se puede utilizar el sensor de orientación para determinar la posición de un dispositivo dentro de un marco de referencia.

El sensor de campo magnético terrestre y el sensor de orientación proporcionan matrices tridimensionales de valores para cada evento sensor. El sensor de campo magnético proporciona los valores de intensidad de campo para cada uno de los tres ejes de coordenadas durante un único evento del sensor. De igual forma, el sensor de orientación proporciona los grados de rotación para cada eje en una matriz de tres dimensiones. Por el contrario, el sensor de proximidad proporciona un único valor para cada evento sensor (4).

2.4.3. GIROSCOPIO

Un giroscopio es un dispositivo mecánico constituido por un disco circular simétrico, que puede rotar libremente sobre cualquier eje libre situado en su centro de gravedad, manteniendo inmóvil su eje de rotación (6). Sirve para medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo.

Este sensor permite medir y mantener la orientación de un teléfono móvil pudiendo cambiar y controlar la orientación, la dirección, el movimiento angular y la rotación de un dispositivo. En un teléfono móvil, pueden determinar la posición y orientación del mismo; de igual forma pueden llevar a cabo funciones de reconocimiento de gestos.

Su funcionamiento está basado en la conservación del momento angular, por eso es utilizado para medir la orientación o para mantenerla haciendo uso de las fuerzas que ejercen en su sistema de balanceo. Se caracteriza fundamentalmente por dos propiedades: la inercia giroscópica y la precesión. Estas propiedades se manifiestan en todos los cuerpos en rotación, incluida la Tierra.

Los Smartphones actuales, incluyen nuevas funciones que requieren el uso de estos sensores para cambiar la orientación de la pantalla del dispositivo automáticamente. Lo normal es usar el giroscopio junto con el acelerómetro para obtener así medidas mucho más precisas, que junto con el sistema operativo del

dispositivo o con las diferentes aplicaciones software, pueden llevar a cabo gran cantidad de acciones.

2.4.4. GPS (*Global Positioning System*)

Se trata de un sistema de posicionamiento que recibe señales de diferentes satélites. Actualmente el sistema de GPS cuenta con un total de 24 satélites en órbita sobre la Tierra, todos se mueven de una forma específica y sincronizada para lograr cubrir toda la superficie terrestre (7).

Para que un dispositivo pueda determinar su posición geográfica, necesita como mínimo la recepción de tres satélites. De cada uno de ellos obtiene su identificación y hora. En base a esos datos, el dispositivo sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al dispositivo y con ello la distancia a dicho satélite mediante triangulación. Una vez conocidas las posiciones de cada uno de los satélites resulta fácil conocer la posición absoluta o coordenadas realiza del punto de medición.

Hoy en día cada vez es más habitual que cualquier dispositivo móvil que llevemos haga uso de este sistema incorporando en su hardware un GPS. Las aplicaciones que se le dan a este sistema son cada vez mayores, desde situarnos en un mapa hasta recomendarnos lugares según criterios que se encuentren cerca de nosotros.

Aunque este sistema parece que funciona bajo cualquier circunstancia, lo cierto es que cuenta con limitaciones debido a lugares en los que las señales que se reciben son defectuosas y llegan con muy poca intensidad, como pueden ser túneles o espacios rodeados de una gran cantidad de edificios. Por ello, últimamente en muchos dispositivos se usa como respaldo otro sistema de posicionamiento que cuenta en órbita con 24 satélites, se trata de GLONASS y se trata de un sistema de navegación desarrollado por los soviéticos. No son dos sistemas independientes, ambos cooperan para poder situar los dispositivos cuando es necesario.

2.5. APLICACIONES ACTUALES

En la actualidad los sensores son usados en multitud de campos teniendo usos tan diferentes como los que se comentan a continuación.

2.5.1. MOVES

Se trata de una aplicación móvil tanto para iOS como para Android que hace uso de los sensores de los dispositivos móviles para grabar tus movimientos detectando si te encuentras caminando, montando en bici o corriendo. En la interfaz que presenta se puede observar tanto los pasos que has dado como la duración y la distancia recorrida además de realizar un cálculo aproximado de las calorías consumidas.

La aplicación se encuentra continuamente en ejecución tomando los datos de los diferentes sensores pero si el usuario lo desea, puede parar su ejecución para que la batería de su dispositivo no se agote rápidamente. En las especificaciones de la aplicación se comenta que no debería consumir mucha batería, pero aun así recomiendan que el dispositivo se cargue todas las noches (8).

2.5.2. CAMISETAS INTELIGENTES

Se trata de camisetas equipadas con diversos sensores que capturan, entre otras, la actividad respiratoria y del corazón del usuario, así como su temperatura corporal. Están diseñadas principalmente para personas con enfermedad obstructiva crónica o con insuficiencia renal crónica, pero también pueden ser muy útiles para deportistas de alto nivel para monitorear su actividad durante sus entrenamientos.

Los datos recogidos por estos sensores son enviados a un dispositivo móvil como un Smartphone o PDA, que mediante un software inteligente, analiza los datos recibidos y los envía al asistente médico en cuestión (9).

2.5.3. JUEGOS

Hoy en día la mayor parte de los juegos que podemos encontrar en los dispositivos móviles funcionan gracias a la interacción por parte del usuario con el móvil moviéndolo directamente, normalmente realizando movimientos a la izquierda o a la derecha.

Debido a estos movimientos, en el juego se producen reacciones según su velocidad o tipo. Estos valores son recogidos por los sensores que se encuentran en los dispositivos y que son interpretados por la aplicación instalada en cuestión.

Para este tipo de aplicaciones los sensores que se leen suelen ser tanto el acelerómetro como el giroscopio.

2.5.4. EFICIENCIA DE RECURSOS EN DISPOSITIVOS

Uno de los usos que se hace de los sensores actualmente es el de mejorar el rendimiento de los dispositivos y de los recursos que utilizan para mejorar entre otras cosas su autonomía.

El último dispositivo fabricado por Apple, el iPhone 5S, está equipado con dos procesadores, uno de ellos se encarga únicamente de la lectura de los sensores, y el otro del resto de tareas necesarias para el correcto funcionamiento del aparato.

El procesador que se encarga de leer los datos de los sensores, llamado M7, es capaz de detectar cuando el individuo se encuentra caminando o cuando se desplaza en coche para inhabilitar ciertas funciones del equipo o modificar su comportamiento. Por ejemplo, si el individuo se encuentra desplazándose en coche, el dispositivo deja de buscar continuamente puntos WiFi para conectarse. Este procesador también detecta cuando el dispositivo lleva mucho tiempo sin moverse, por ejemplo cuando el usuario está durmiendo, para reducir el consumo de la batería (10).

3. ANÁLISIS

Una vez realizada la introducción sobre la temática del proyecto y estudiados los antecedentes, en los siguientes subapartados se exponen los diferentes casos de uso identificados en los sistemas, así como los requisitos de software que deben cumplir y la relación existente entre estos dos. Por último se comentará el entorno y tecnología necesaria para desarrollar los sistemas.

El trabajo descrito en este capítulo servirá como base para el siguiente de diseño de las aplicaciones.

3.4. CASOS DE USO

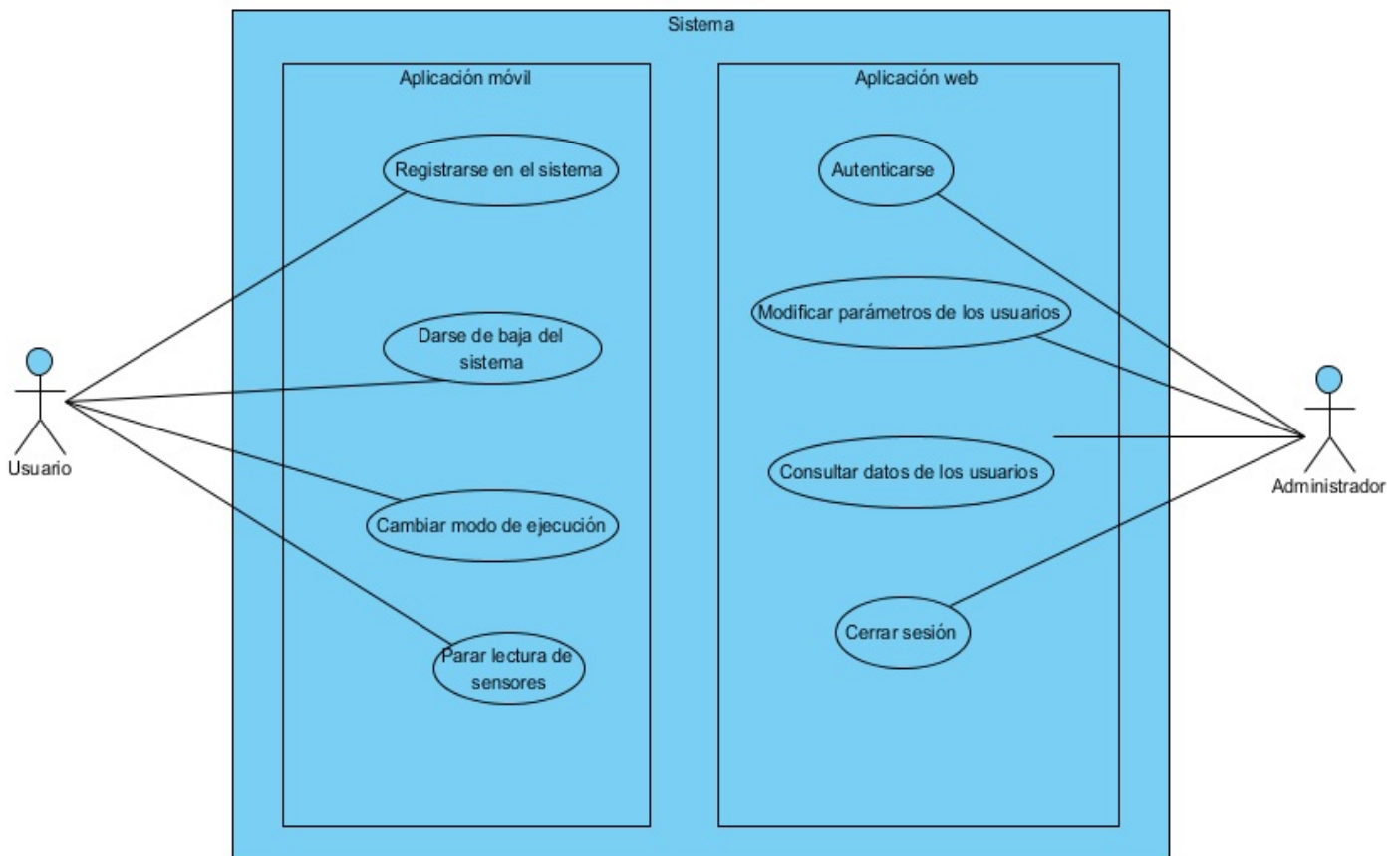


Imagen 3. Diagrama de casos de uso.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN TELÉFONOS INTELIGENTES

Para el desarrollo de la aplicación se diferencian claramente dos sistemas, el perteneciente a la aplicación web y el perteneciente a la aplicación móvil. Además existen dos actores diferentes, cada uno de ellos interactúa con el sistema que le pertenece, siendo el actor “usuario” el que interactúa con la aplicación móvil y el actor “administrador” el que interactúa con la aplicación web.

El actor “administrador” se encarga de gestionar los datos del actor “usuario” en el servidor, así como realizar modificaciones en las aplicaciones de los usuarios que se encuentren registrados en el sistema. Necesita tener asignado un nombre de usuario así como una contraseña para poder autenticarse en la aplicación web y acceder a la consola de administración.

El actor “usuario” es aquel que tiene instalada la aplicación en su dispositivo y se encuentra registrado en la base de datos del servidor de la aplicación web. Su función es controlar los modos de ejecución de la aplicación móvil para que se encargue de recoger datos y enviarlos al servidor automáticamente.

Es importante destacar que cualquiera puede ser un actor “usuario” mientras que únicamente hay una entidad “administrador”.

Identificador	CU-01
Título	Registrarse en el sistema
Descripción	El usuario podrá registrarse en el sistema para que los datos recogidos por los sensores puedan enviarse al servidor.
Precondiciones	El usuario no debe estar registrado en el momento de realizar la acción.
Flujo principal	1.- El usuario accede a la aplicación. 2.- Elige la opción “sign up”. 3.- El sistema realiza el registro del usuario en la aplicación web.
Flujo alternativo	-----

Tabla 2. Caso de uso 01.

Identificador	CU-02
Título	Darse de baja en el sistema
Descripción	El usuario se puede dar de baja en el sistema siempre que quiera.
Precondiciones	Debe estar registrado en el momento de realizar la acción.
Flujo principal	1.- El usuario accede a la aplicación. 2.- Por medio de las opciones de la aplicación elige la opción "logout". 3.- El sistema pide confirmación. 4.- El usuario confirma la acción.
Flujo alternativo	3.1.- El usuario cancela la acción. 4.1.- El sistema pregunta si se quieren borrar los datos recogidos almacenados en la aplicación web. 4.2.- El usuario acepta la opción. 4.3.- El sistema borra todos sus datos almacenados en la aplicación web.

Tabla 3. Caso de uso 02

Identificador	CU-03
Título	Cambiar modo de ejecución.
Descripción	El usuario podrá cambiar el modo de ejecución de la aplicación eligiendo entre modo "car", "walk" o "standby".
Precondiciones	1.- El usuario debe estar registrado en el sistema. 2.- La aplicación debe estar en ejecución. 3.- Debe estar en modo "general".
Flujo principal	1.- El usuario pulsa un modo de ejecución. 2.- El sistema para el modo de ejecución actual, y comienza en nuevo. Muestra por pantalla el modo actual de ejecución.
Flujo alternativo	1.1.- Si se encontraba ya en otro modo, debe terminar ese modo. 1.2.- Elegir el modo deseado.

Tabla 4. Caso de uso 03.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

Identificador	CU-04
Título	Parar lectura de sensores.
Descripción	El usuario podrá detener la lectura de datos de los sensores por la aplicación cuando desee.
Precondiciones	1.- El usuario debe estar registrado en el sistema. 2.- La aplicación debe estar en ejecución.
Flujo principal	1.- El usuario elige la opción “stop”. 2.- El sistema deja de leer los datos recogidos por los sensores.
Flujo alternativo	-----

Tabla 5. Caso de uso 04.

Identificador	CU-05
Título	Autenticarse
Descripción	El administrador debe autenticarse para acceder a la administración del sistema.
Precondiciones	Debe estar dado de alta en la base de datos del servidor.
Flujo principal	1.- Accede a la web de administración. 2.- Introduce su nombre de usuario y contraseña. 3.- Si es correcto accede a la consola de administración.
Flujo alternativo	2.1.- Si la información es errónea, se le informa para que vuelva a intentar acceder.

Tabla 5. Caso de uso 05.

Identificador	CU-06
Título	Modificar parámetros de los usuarios.
Descripción	El administrador podrá modificar algunos parámetros de la aplicación de los usuarios desde la consola de administración como el intervalo de envío de los datos recogidos y el intervalo de captura de esos datos.
Precondiciones	Estar autenticado en el sistema.
Flujo principal	1.- Ir a ajustes. 2.- Elegir los parámetros a modificar. 3.- El sistema pide confirmación de la acción. 4.- El administrador confirma la acción.
Flujo alternativo	-----

Tabla 6. Caso de uso 06

Identificador	CU-07
Título	Consultar datos de los usuarios.
Descripción	El administrador podrá realizar consultas de datos a partir de un formulario.
Precondiciones	Estar autenticado en el sistema.
Flujo principal	1.- Ir a la sección "Data". 2.- Elegir los criterios de búsqueda. 3.- Pulsar "Search". 3.- El sistema realiza la búsqueda de coincidencias en la base de datos. 4- El sistema muestra los resultados en una tabla.
Flujo alternativo	5.- El administrador elige la opción de "Export data". 6.- El sistema vuelca los datos a un fichero y se descarga al sistema de ficheros del administrador.

Tabla 7. Caso de uso 07.

Identificador	CU-08
Título	Cerrar sesión.
Descripción	El administrador podrá cerrar su sesión activa.
Precondiciones	Estar autenticarse en el sistema.
Flujo principal	1.- El administrador en la pantalla de administración elige cerrar su sesión. 3.- El sistema redirecciona al administrador a la pantalla de autenticarse eliminando previamente sus sesión activa.
Flujo alternativo	-----

Tabla 8. Caso de uso 08.

3.5. REQUISITOS DE SOFTWARE

Se trata de los requisitos que debe cumplir el sistema desarrollado para que cumpla de manera correcta las expectativas. Éstos los podemos clasificar en requisitos funcionales y no funcionales.

Los requisitos de software no funcionales se refieren a aquellos que no describen una acción que puede llevar a cabo el software sino que se refieren a temas de rendimiento, seguridad, concurrencia, escalabilidad...etc.

Los requisitos de software funcionales son los que definen el comportamiento interno del software, establecen los comportamientos del sistema como detalles técnicos, manipulación de datos y otras funcionalidades específicas como los casos de uso llevados a la práctica.

La clasificación que se llevará a cabo es la siguiente:

- Identificador: Si es funcional RSF-XX, si es no funcional RSNF-XX
 - XX: Número de requisito.
- Descripción: Explicación breve del requisito.
- Prioridad: Alta, media o baja según la importancia para ser desarrollado por el equipo.
- Verificabilidad: Prueba que demuestra su correcto funcionamiento.

3.5.1. REQUISITOS FUNCIONALES

Identificador	RSF-01
Descripción	El software de la aplicación móvil deberá poder dar de alta a los usuarios en la base de datos del servidor a través de http.
Prioridad	Media.
Verificabilidad	PR-01.

Tabla 9. Requisito de Software Funcional 01.

Identificador	RSF-02
Descripción	El software de la aplicación móvil deberá poder dar de alta a los usuarios en el servicio de GCM (Google Cloud Messaging) a través de Google Cloud Endpoints.
Prioridad	Media.
Verificabilidad	PR-01.

Tabla 10. Requisito de Software Funcional 02.

Identificador	RSF-03
Descripción	El software de la aplicación móvil permitirá dar de baja del sistema a los usuarios eliminándolos de la base de datos.
Prioridad	Baja.
Verificabilidad	PR-02.

Tabla 11. Requisito de Software Funcional 03.

Identificador	RSF-04
Descripción	El software de la aplicación móvil permitirá iniciar la lectura de datos de los sensores de posición, aceleración, giroscopio y GPS si la aplicación se encuentra en modo "inactive".
Prioridad	Media.
Verificabilidad	PR-04.

Tabla 12. Requisito de Software Funcional 04.

Identificador	RSF-05
Descripción	El software de la aplicación móvil permitirá cambiar el modo de ejecución eligiendo entre los modos "walk", "car" y "standby".
Prioridad	Alta.
Verificabilidad	PR-05, PR-06.

Tabla 13. Requisito de Software Funcional 05.

Identificador	RSF-06
Descripción	El software de la aplicación móvil permitirá parar la lectura de los sensores.
Prioridad	Media.
Verificabilidad	PR-07.

Tabla 14. Requisito de Software Funcional 06.

Identificador	RSF-07
Descripción	El software de la aplicación móvil permitirá bajo petición del usuario borrar los datos recogidos por sus sensores del servidor.
Prioridad	Media.
Verificabilidad	PR-03.

Tabla 15. Requisito de Software Funcional 07.

Identificador	RSF-08
Descripción	El software de la aplicación móvil dará la opción de activar el GPS si se encuentra desactivado.
Prioridad	Media.
Verificabilidad	PR-08.

Tabla 16. Requisito de Software Funcional 08.

Identificador	RSF-09
Descripción	El software de la aplicación web permitirá acceder al sistema con un usuario y contraseña.
Prioridad	Alta.
Verificabilidad	PR-09.

Tabla 17. Requisito de Software Funcional 09.

Identificador	RSF-10
Descripción	El software de la aplicación web permitirá cerrar sesión iniciada por el administrador, quedando todas las variables de sesión eliminadas.
Prioridad	Alta.
Verificabilidad	PR-11.

Tabla 18. Requisito de Software Funcional 10.

Identificador	RSF-11
Descripción	El software de la aplicación web permitirá consultar los datos de los sensores recogidos por la aplicación web de un usuario introduciendo su identificador generado en el datastore y activando los <i>checkbox</i> de los sensores que desea obtener los datos.
Prioridad	Media.
Verificabilidad	PR-12.

Tabla 19. Requisito de Software Funcional 11.

Identificador	RSF-12
Descripción	El software de la aplicación web permitirá modificar el intervalo de envío de datos al servidor así como el intervalo de captura de estos datos por los dispositivos de los clientes registrados.
Prioridad	Media.
Verificabilidad	PR-13.

Tabla 20. Requisito de Software Funcional 12.

Identificador	RSF-13
Descripción	El software de la aplicación web permitirá descargar los datos consultados de un usuario en un fichero.
Prioridad	Baja.
Verificabilidad	PR-14.

Tabla 21. Requisito de Software Funcional 13.

Identificador	RSF-14
Descripción	El software de la aplicación web en el caso de que se intente autenticar un usuario que no está dado de alta en el sistema, mostrará un error.
Prioridad	Alta.
Verificabilidad	PR-10.

Tabla 22. Requisito de Software Funcional 14.

Identificador	RSF-15
Descripción	El software de la aplicación web en el caso de introducir valores incorrectos al intentar modificar parámetros de los clientes mostrará un error.
Prioridad	Alta.
Verificabilidad	PR-15 y PR-16.

Tabla 23. Requisito de Software Funcional 15.

Identificador	RSF-16
Descripción	El software de la aplicación web en el caso de que se introduzca un valor no numérico al consultar los datos de un usuario, no realizará ninguna búsqueda.
Prioridad	Media.
Verificabilidad	PR-17.

Tabla 24. Requisito de Software Funcional 16.

Identificador	RSF-17
Descripción	El software de la aplicación móvil será capaz de realizar el registro de los sensores de aceleración, giroscopio y posición.
Prioridad	Alta.
Verificabilidad	PR-18.

Tabla 25. Requisito de Software Funcional 17.

Identificador	RSF-18
Descripción	El software de la aplicación móvil será capaz de leer los datos de los sensores de aceleración giroscopio y posición, una vez registrados con una periodicidad indicada por el intervalo de captura definido la primera vez que se inicia la aplicación.
Prioridad	Alta.
Verificabilidad	PR-18.

Tabla 26. Requisito de Software Funcional 18.

Identificador	RSF-19
Descripción	El software de la aplicación móvil será capaz de enviar los datos recogidos al servidor con una periodicidad indicada por el intervalo de envío definido la primera vez que se inicia la aplicación.
Prioridad	Media.
Verificabilidad	PR-18.

Tabla 27. Requisito de Software Funcional 19.

Identificador	RSF-20
Descripción	El software de la aplicación móvil será capaz de leer los datos del GPS si éste se encuentra activado y cada vez que cambie la posición del usuario.
Prioridad	Alta.
Verificabilidad	PR-18.

Tabla 28. Requisito de Software Funcional 20.

3.5.2. REQUISITOS NO FUNCIONALES

Identificador	RSNF-01
Descripción	La aplicación móvil realizará el envío de datos de los sensores a través de <i>Google Cloud Endpoints</i> .
Prioridad	Media.

Tabla 29. Requisito de Software Funcional 01.

Identificador	RSNF-02
Descripción	La aplicación web se comunicará con la aplicación móvil para la modificación del intervalo de captura y de envío de datos a través de <i>Google Cloud Messaging</i> .
Prioridad	Media.

Tabla 30. Requisito de Software Funcional 02.

Identificador	RSNF-03
Descripción	Los procesos de registro, autenticación y transmisión de datos se realizarán con el protocolo SSL (12).
Prioridad	Media.

Tabla 31. Requisito de Software no Funcional 03.

Identificador	RSNF-04
Descripción	El acceso a la aplicación web se realizará por un único usuario y con una sesión que expira a los 5 minutos de inactividad.
Prioridad	Baja.

Tabla 32. Requisito de Software no Funcional 04.

Identificador	RSNF-05
Descripción	Las interfaces desarrolladas para la aplicación web y móvil deberán estar en inglés.
Prioridad	Media.

Tabla 33. Requisito de Software no Funcional 05.

Identificador	RSNF-06
Descripción	Las unidades de medida obtenidas para cada uno de los sensores será el siguiente: Acelerómetro: m/s ² . Giroscopio: radianes/s. Posición: radianes. GPS: grados.
Prioridad	Media.

Tabla 34. Requisito de Software no Funcional 06.

3.6. TRAZABILIDAD DE REQUISITOS DE SOFTWARE Y CASOS DE USO

En la siguiente matriz se recoge la correspondencia de cada uno de los casos de uso identificados en los sistemas con los requisitos de software que tienen en cuenta esa funcionalidad.

		Casos de uso							
		01	02	03	04	05	06	07	08
Requisitos de software	01	X							
	02	X							
	03		X						
	04								
	05			X					
	06				X				
	07								
	08								
	09					X			
	10								X
	11							X	
	12						X		
	13							X	
	14					X			
	15						X		
	16							X	
	17	X							
	18			X					
	19			X					
	20			X					

Tabla 35. Trazabilidad requisitos de software y casos de uso.

3.7. ENTORNO DE DESARROLLO

En las siguientes subsecciones se explicarán las herramientas que han sido necesarias para realizar el desarrollo tanto de la aplicación móvil como web.

3.7.1. JAVA

Se trata de un lenguaje de programación orientado a objetos que fue desarrollado originalmente por James Gosling de Sun Microsystems, actualmente Oracle es su propietaria (10). La característica más llamativa de las aplicaciones desarrolladas en este lenguaje es que son portables y pueden ejecutarse en cualquier máquina sin importar su arquitectura, esto permite que un desarrollador codifique el programa una vez y pueda ejecutarlo las veces que quiera donde sea necesario sin preocuparse de realizar cambios para adecuarlo a la arquitectura de la nueva máquina.

3.7.2. BOOTSTRAP

Es una herramienta de trabajo para crear aplicaciones web. Contiene ayudas de HTML y CSS así como plantillas para la realización de formularios, tablas y barras de navegación.

El *framework* es muy reciente ya que se comenzó a desarrollar por Twitter en el año 2010 y ya ha sido usado por organizaciones tan importantes como la NASA (11).

3.7.3. ECLIPSE IDE

Entorno de programación multiplataforma para desarrollar aplicaciones en diversos lenguajes. Fue desarrollado por IBM, pero actualmente es desarrollado por la Fundación Eclipse, la cual no tiene ánimo de lucro y promueve el uso de código abierto.

Este entorno de desarrollo cuenta con una gran cantidad de *plugins* para realizar diferentes tareas como por ejemplo para disponer de un repositorio de código

para el mantenimiento de versiones y control de código (Egit) o *plugins* para el desarrollo de aplicaciones Android (ADT). (12)

3.8. TECNOLOGÍA

En los siguientes subapartados se describirán las diferentes tecnologías sobre las que se apoya y hace uso la aplicación para su correcto funcionamiento y comunicación entre componentes.

3.8.1. GOOGLE APP ENGINE

Se trata de un alojamiento gratuito, con límite de cuota, para aplicaciones web. Actualmente soporta los lenguajes de programación Java, Python, Go y PHP. El servicio fue lanzado en el año 2008 (13). El SDK que se proporciona para desarrollar aplicaciones en esta plataforma permite que se pueda ejecutar la aplicación en local para realizar las pruebas y de esta manera no realizar la subida de la aplicación a la plataforma con posibles errores.

A diferencia de otros almacenes de aplicaciones, *Google App Engine*, no proporciona una base de datos relacional del tipo MySQL, sino que pone a disposición del desarrollador un almacén de datos llamado *datastore* orientado a objetos ofreciendo un gran escalado en las aplicaciones desarrolladas.

3.8.2. GOOGLE CLOUD ENDPOINTS

Se trata de un conjunto de herramientas y librerías que permiten de forma sencilla realizar un API y conjunto de librerías para comunicar diferentes clientes con *Google App Engine*. Esta tecnología facilita el desarrollo de aplicaciones móviles que deben comunicarse con un servidor web para el intercambio de información; es compatible tanto para clientes con el sistema operativo Android como con el sistema operativo iOS.

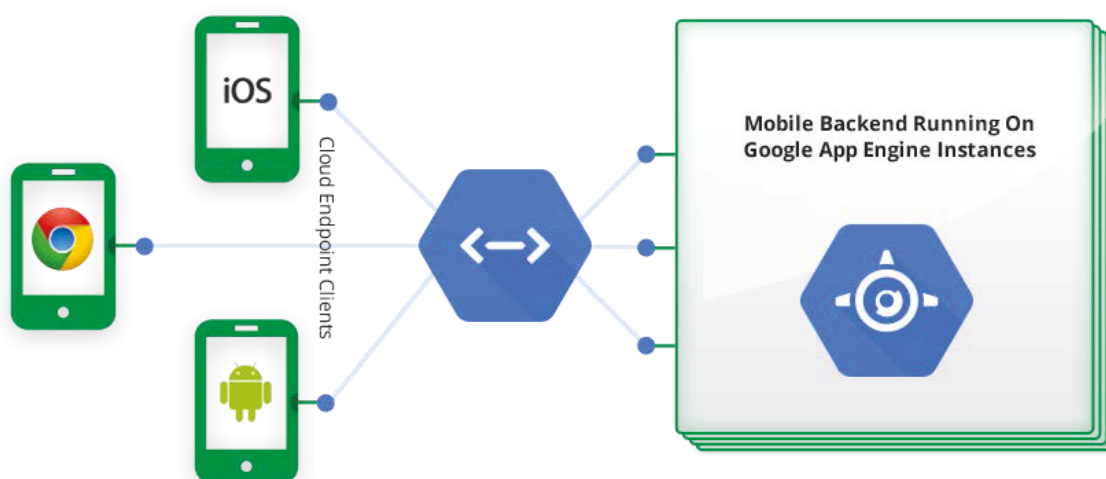


Imagen 4. Arquitectura de *Google Cloud Endpoints* (14).

3.8.3. PLATAFORMA

Hoy en día, el mercado de los dispositivos móviles está gobernado principalmente por dos sistemas operativos: Android el cual está desarrollado por Google e iOS que está desarrollado por Apple. Existen otros sistemas operativos que en los últimos tiempos están experimentando un crecimiento, pero aun así se encuentran muy alejados con respecto a iOS y Android.

El sistema operativo de Google se encuentra presente en una gran cantidad de Smartphones de diferentes marcas; éstas se encargan de realizar modificaciones respecto a la versión base desarrollada por Google. iOS, sin embargo, solo se encuentra presente en los Smartphones fabricado por Apple, el iPhone.

La consecuencia de que Android se encuentre presente en diferentes fabricantes de Smartphones es la gran cantidad de versiones que se pueden encontrar hoy en día en el mercado debido a que son ellos los que deciden si actualizan sus dispositivos a la última versión o no.

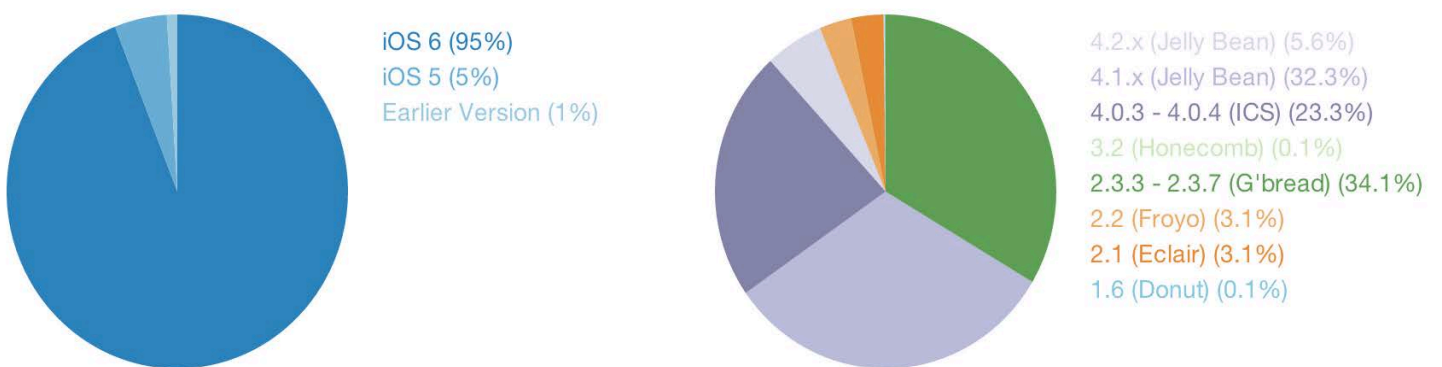


Imagen 5. Comparación de versiones iOS vs Android (15).

Debido a todo esto, Android ya se encuentra presente en el 80% del total de dispositivos móviles del mercado llegando así a una mayor cantidad de usuarios que iOS.

Sistema Operativo	Volumen de ventas	Cuota de mercado
Android	211.6	81.0%
iOS	33.8	12.9%
Windows Phone	9.5	3.6%
BlackBerry	4.5	1.7%
Otros	1.7	0.6%
Total	261.1	100.0%

Tabla 36. Top 4 de sistemas operativos 3º trimestre 2013 (unidades en millones) (16).

3.5.4. OTRAS TECNOLOGÍAS

Aunque la tecnología usada para el alojamiento de la aplicación web ha sido *Google App Engine*, existían otras alternativas para cubrir esta función. A continuación se van a comentar algunas según una clasificación establecida, y los pros y contras de cada una y el porqué de la elección.

Podemos diferenciar entre plataformas propias como puede ser Apache, o *Cloud* como Amazon y *Google App Engine*. La característica principal que se buscaba es que la plataforma elegida generara poca carga de trabajo en el desarrollo del proyecto, y evitar el tener que configurar muchos parámetros para el correcto funcionamiento de la aplicación web.

Por la razón anterior se desechó la opción de la plataforma Apache ya que hay que realizar muchas configuraciones del servidor para poder usarlo de manera correcta y eficiente, además de que no es tan fácil de integrar en otras plataformas como sí lo son Amazon y *Google App Engine*. Entre las plataformas Cloud, ambas son escalables según las necesidades, factor a tener muy en cuenta para que pueda ser accedida por una gran cantidad de usuarios y que no se produzcan por ejemplo errores de acceso a las bases de datos, pero en el servicio ofrecido por Amazon también hay que realizar configuraciones, lo que supone dedicarle tiempo a esa tarea y no a otras más importantes.

Google App Engine ha sido la opción elegida principalmente por el gran API de ayuda para desarrollo de aplicaciones que ofrece y la gran autoescalabilidad sin tener que configurar nada ni tener que estar pendiente de que funciona correctamente; únicamente hay que realizar la aplicación y añadirla a la plataforma web.

3.9. DIAGRAMA DE ESTADOS

Llegados a este punto del análisis, en la aplicación móvil se identifican diferentes estados que son el resultado de la interacción del usuario con la aplicación.

Los estados diferenciados se basan en el tipo de ejecución en el que se puede encontrar la aplicación según su modo. Hay cuatro modos de ejecución, a saber: “car”, “walk”, “standby” y “general”; cuando no se encuentra en ninguno de estos modos, la aplicación está inactiva.

Se ha decidido juntar los modos de ejecución “car”, “walk” y “standby” en uno solo llamado “específico”; esto se debe a que a cada uno de ellos se llega de la misma forma (pulsando su botón correspondiente) y para salir de ellos e iniciar otro igual. El modo de ejecución general forma un solo estado llamado “general”. Cuando la aplicación no se encuentra en ninguno de esos modos está inactiva por lo que se identifica otro estado. Por lo tanto, son tres estados: general, específico e inactivo.

A continuación se muestra el diagrama de estados correspondiente a la aplicación móvil una vez realizado este análisis.

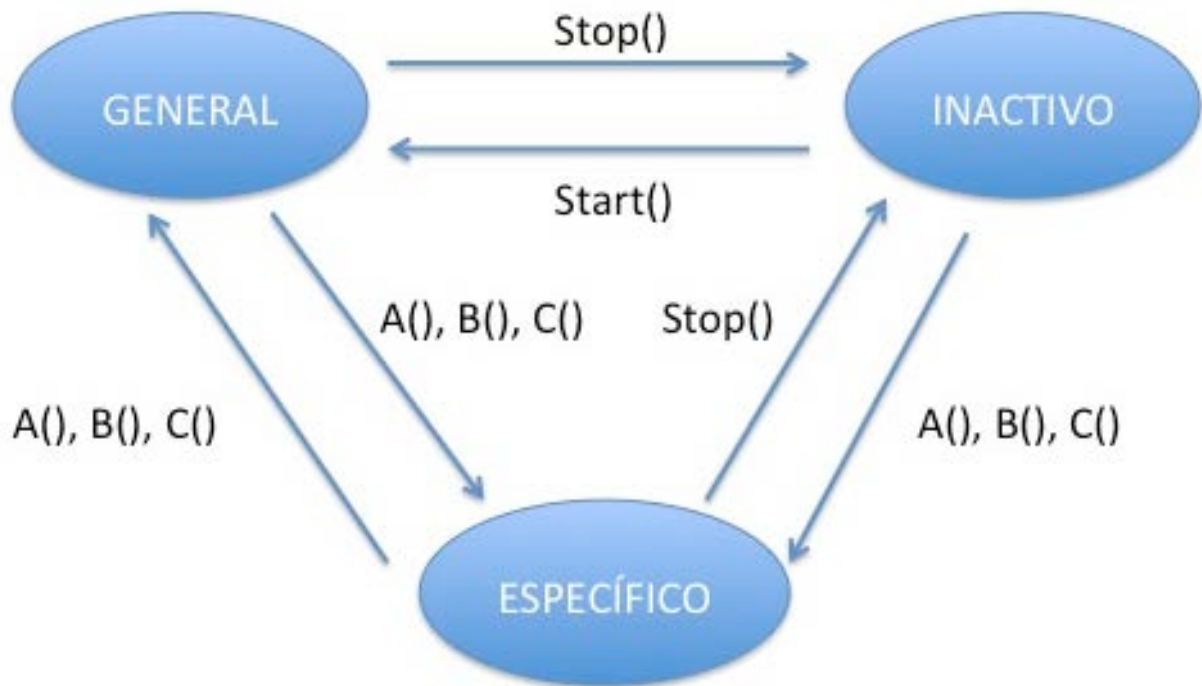


Imagen 6. Diagrama de estados aplicación móvil.

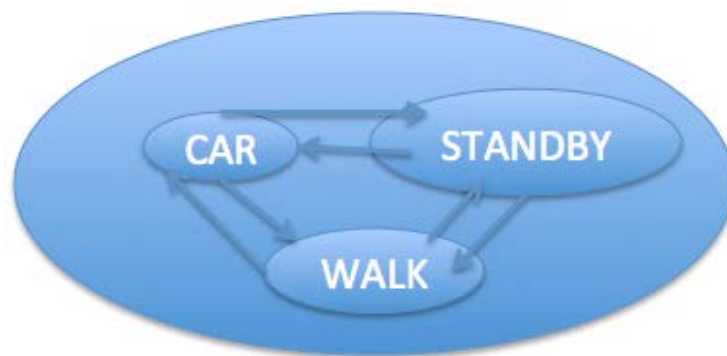


Imagen 7. Zoom de los modos contenidos en el estado Específico.

Como se observa en el diagrama, hay tres estados posibles en los que la aplicación móvil se puede encontrar: estado general, inactivo y específico. En el estado específico, la aplicación se puede encontrar en los modos “car”, “walk” o “standby”.

Para pasar de un estado a otro se requiere la acción del usuario, y para ello debe pulsar un botón dentro de la aplicación. Esto provoca las transiciones entre estados. Por ejemplo, para pasar de estado general a inactivo, el usuario debería pulsar el botón “stop” de la aplicación. Si por el contrario quisiera pasar del estado inactivo al general debería pulsar el botón “start”.

Las acciones A, B y C con que se puede pasar de estado “general” a “específico” y viceversa, se refieren a diferentes botones de la aplicación correspondientes a los modos “car”, “walk” y “standby”. Por ejemplo, estando en el modo de ejecución “general”, si quisiéramos pasar al modo de ejecución “walk”, deberíamos pulsar el botón correspondiente para dicho modo que es diferente al botón del modo “car” y “standby”. Para pasar del modo “walk” al “general”, el procedimiento sería el mismo; y así, para el resto de modos dentro del estado específico.

La información del diagrama con los diferentes estados iniciales y finales con sus correspondientes transiciones se encuentran recogidas en la siguiente tabla:

Estado Inicial	Estado Final	Botón pulsado
General	Inactivo	Stop
Inactivo	General	Start
Específico_X	General	X
Específico_X	Específico_Y	**
Inactivo	Específico_X	X
General	Específico_X	X
Específico_X	Inactivo	Stop

Tabla 37. Transiciones y estados de la aplicación móvil.

Como se ha comentado anteriormente, el estado específico puede ser modo “walk”, “standby” o “car”.

- Específico X: se refiere al estado de la aplicación correspondiente con el modo de ejecución “car”, “walk” o “standby”.
- Específico Y: se refiere al estado de la aplicación correspondiente con el modo de ejecución “car”, “walk” o “standby” diferente de Específico_X. Por ejemplo, si el estado inicial es “Específico_car”, el estado final podría ser “Específico_walk”, o “Específico_stamdby” pero nunca “Específico_car”.

** No se puede realizar la transición, antes debe pasar por el estado general o inactivo.

3.10. CATÁLOGO DE PRUEBAS

En esta sección se presenta el conjunto de pruebas a realizar una vez se encuentren tanto la aplicación web como la aplicación móvil implementadas para verificar que el software desarrollado es de calidad cumpliendo todos los requisitos descritos en la sección 3.5.

Cada prueba descrita se compone de los siguientes campos:

- Identificador: Cadena de texto compuesta por PR-XX que identifica unívocamente a cada una de las pruebas donde XX se refiere a al número de prueba. Por ejemplo, la prueba número uno sería PR-01.
- Título: Resumen característico de la prueba.
- Descripción: Explicación detallada de la prueba y los pasos a seguir para llevarla a cabo.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN TELÉFONOS INTELIGENTES

- Parámetros de entrada: Valores a introducir en el sistema para realizar la prueba. En el caso de que no se requiera ninguno, el valor para este campo será N/A (no aplica).
- Salida: Resultado que se debe obtener para que la prueba se considere satisfactoria.

Todo el conjunto de pruebas será realizado cada quince días durante toda la vida de los sistemas para garantizar que su funcionamiento es el correcto rellenando la siguiente plantilla.

PRUEBA	RESULTADO
PR-01	
PR_02	
PR-03	
....	

Tabla 38. Checklist de resultados de pruebas.

En el caso de que la prueba se considere satisfactoria, el valor de resultado será “OK”; en el caso contrario, será “KO”.

Identificador	PR-01
Título	Alta automática en el sistema.
Descripción	Se instalará la aplicación en un dispositivo móvil con el sistema operativo Android y se esperará a que se inicie la aplicación.
Parámetros de entrada	N/A.
Salida	La aplicación móvil comienza a realizar la lectura de los sensores en modo “general”. En la base de datos del servidor se encuentra dado de alta el usuario en la tabla “Usuarios”.

Tabla 39. Alta automática en el sistema.

Identificador	PR-02
Título	Baja en el sistema.
Descripción	Una vez se encuentra la aplicación instalada, se accederá dentro de la aplicación al menú de opciones y se pulsará “logout” y confirmará la acción en el primer cuadro de diálogo mostrado y rechazará el segundo.
Parámetros de entrada	N/A.
Salida	La aplicación mostrará una nueva pantalla pidiendo que el usuario se registre de nuevo en el sistema en caso de que quiera volver a usar la aplicación. En la base de datos del servidor se ha borrado al usuario.

Tabla 40. Baja en el sistema.

Identificador	PR-03
Título	Borrado de datos del sistema.
Descripción	Para la realización de esta prueba, se debe realizar la PR-02 confirmando también el segundo cuadro de diálogo.
Parámetros de entrada	N/A.
Salida	Además de la salida de la PR-02, en la tabla “Mensajes” de la base de datos del servidor, se habrán eliminado todos los pertenecientes a dicho usuario.

Tabla 41. Borrado de datos del sistema.

Identificador	PR-04
Título	Iniciar lectura de sensores.
Descripción	Con la aplicación abierta, se pulsará el botón “stop” entrando en el modo “inactive”; una vez en ese modo, se pulsará “start”.
Parámetros de entrada	N/A.
Salida	La aplicación comenzará a leer los datos de los sensores, reclamará la activación del GPS en caso de que se encuentre desactivado, y mostrará en la pantalla el mensaje “general mode activated”. El botón “start” se cambiará a color rojo.

Tabla 42. Iniciar lectura de sensores.

Identificador	PR-05
Título	Cambiar modo de ejecución.
Descripción	Con la aplicación iniciada en modo “general”, se pulsará el botón “car”.
Parámetros de entrada	N/A.
Salida	La aplicación cambia a modo “car”. Se muestra por pantalla “car mode activated” y el botón pulsado se cambia a color rojo.

Tabla 43. Cambiar modo de ejecución.

Identificador	PR-06
Título	Cambio de modo no permitido.
Descripción	Con la aplicación iniciada en modo “general”, se pulsará el botón “car” y a continuación el botón “walk”.
Parámetros de entrada	N/A.
Salida	La aplicación mostrará el mensaje “Must finish car mode to start another”.

Tabla 44. Cambio de modo no permitido.

Identificador	PR-07
Título	Detener lectura de sensores.
Descripción	Con la aplicación iniciada, se pulsará el botón “stop”.
Parámetros de entrada	N/A.
Salida	La aplicación se queda en modo “inactive” mostrando por pantalla “inactive mode activated: to start Reading sensors data click start.” El botón pulsado ahora muestra el texto “start” y en color azul.

Tabla 45. Detener lectura de sensores.

Identificador	PR-08
Título	Pedir activación de GPS.
Descripción	Se desactivará el GPS en el dispositivo en el que se realizará la prueba. A continuación, se abrirá la aplicación.
Parámetros de entrada	N/A.
Salida	La aplicación muestra un cuadro de diálogo dando la opción de activar el GPS del dispositivo o no. Si se acepta se accede al panel de ajustes del dispositivo para activarlo; en el caso de cancelar la acción, sigue la ejecución de la aplicación.

Tabla 46. Pedir activación de GPS.

Identificador	PR-09
Título	Acceso a la aplicación web satisfactorio.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña. Una vez cumplimentados los campos se pulsará "Enter".
Parámetros de entrada	User = pepin Password = 1234
Salida	Se accede a la consola de administración de la aplicación web mostrando en la parte superior el nombre del administrador y la fecha del último acceso.

Tabla 47. Acceso a la aplicación web satisfactoria.

Identificador	PR-10
Título	Acceso a la aplicación web erróneo.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña. Una vez cumplimentados los campos se pulsará "Enter".
Parámetros de entrada	User = pepin Password = 12345
Salida	Se muestra el mensaje "Incorrect administrator's data." y se queda en la misma dirección.

Tabla 48. Acceso a la aplicación web erróneo

Identificador	PR-11
Título	Cierre de sesión del administrador.
Descripción	Una vez dentro de la aplicación web, se debe de hacer click en el hipervínculo "Log out".
Parámetros de entrada	N/A.
Salida	El sistema redirige a la pantalla de autenticación en la aplicación.

Tabla 49. Cierre de sesión del administrador.

Identificador	PR-12
Título	Consultar datos de un usuario en la aplicación web.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña y se pulsar "Enter". Una vez dentro del sistema, se accede a la sección "Data". Y se introduce el valor para el campo "User ID" y se marcan los campos "Position", "Accelerometer", "Gyroscope" y "GPS". Por último se pulsa el botón "Search".
Parámetros de entrada	User = pepin Password = 1234 User ID = 5815383571300352
Salida	Se muestra una tabla con todos los resultados obtenidos y los campos pertenecientes a los sensores truncados para que no se extiendan demasiado las filas.

Tabla 50. Consulta de datos de un usuario en la aplicación web.

Identificador	PR-13
Título	Modificar configuración de los clientes.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña. Una vez cumplimentados los campos se pulsará "Enter". Se pulsa en la sección "Settings" y se introducen los datos para los campos a modificar. Por último se pulsa "Confirm" y en el diálogo que genera el sistema se acepta.
Parámetros de entrada	Capture Interval (ms) = 300 Send Interval (hours) = 1
Salida	El sistema mostrará un diálogo con el siguiente mensaje "The configuration change was made successfully".

Tabla 51. Modificación de parámetros.

Identificador	PR-14
Título	Descarga de datos.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña y se pulsar "Enter". Una vez dentro del sistema, se accede a la sección "Data". Y se introduce el valor para el campo "User ID" y se marcan los campos "Position", "Accelerometer", "Gyroscope" y "GPS". Se pulsa el botón "Search". Y el sistema muestra una tabla con los resultados. Por último se pulsa el botón "Export Data".
Parámetros de entrada	User = pepin Password = 1234 User ID = 5815383571300352
Salida	Comenzará la descarga de un archivo con todos los datos en formato CSV.

Tabla 52. Descarga de datos.

Identificador	PR-15
Título	Modificar configuración de los clientes Error_1.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña. Una vez cumplimentados los campos se pulsará "Enter". Se pulsa en la sección "Settings" y se introducen los datos para los campos a modificar. Por último se pulsa "Confirm".
Parámetros de entrada	Capture Interval (ms) =abc
Salida	El sistema mostrará el mensaje "Only numbers are allowed."

Tabla 53. Error en valor de parámetro a modificar I.

Identificador	PR-16
Título	Modificar configuración de los clientes Error_2.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña. Una vez cumplimentados los campos se pulsará "Enter". Se pulsa en la sección "Settings" y se introducen los datos para los campos a modificar. Por último se pulsa "Confirm".
Parámetros de entrada	Capture Interval (ms) = end Interval (hours) =
Salida	El sistema mostrará el mensaje "You must enter a value at least."

Tabla 54. Error en valor de parámetro a modificar II.

Identificador	PR-17
Título	Consulta de datos errónea de un usuario en la aplicación web.
Descripción	Se accede a la dirección https://evaluasensores.appspot.com y se introduce en los campos el usuario y contraseña y se pulsar "Enter". Una vez dentro del sistema, se accede a la sección "Data". Y se introduce el valor para el campo "User ID" y se marcan los campos "Position", "Accelerometer", "Gyroscope" y "GPS". Por último se pulsa el botón "Search".
Parámetros de entrada	User = pepin Password = 1234 User ID = sdf
Salida	El sistema no muestra ningún resultado.

Tabla 55. Error de consulta de datos.

Identificador	PR-18
Título	Comunicación aplicación móvil y web.
Descripción	Se inicia la aplicación móvil y se deja ejecutando 20 segundos. Se accede a la aplicación web y se comprueba la recepción de un mensaje.
Parámetros de entrada	N/A
Salida	La aplicación web muestra el mensaje recibido.

Tabla 56. Comunicación aplicaciones.

4. DISEÑO

En esta fase de desarrollo se parte del trabajo realizado en el análisis para que el equipo de desarrollo pueda implementar la solución propuesta. Para ello se establece la arquitectura que siguen los sistemas y la comunicación entre ellos así como el patrón de diseño que se seguirá en su desarrollo. Se hace uso de una serie de diagramas que permiten especificar la solución adoptada.

En el siguiente capítulo en el que se trata el tema de la implementación, se hará uso de todas las especificaciones del presente capítulo.

4.4. ARQUITECTURA

El trabajo realizado consta de dos sistemas, la aplicación web y la aplicación móvil. La aplicación web se encuentra alojada en un servidor de aplicaciones Google App Engine y accesible desde un navegador web, mientras que la aplicación móvil se encuentra instalada en dispositivos móviles con sistema operativo Android.

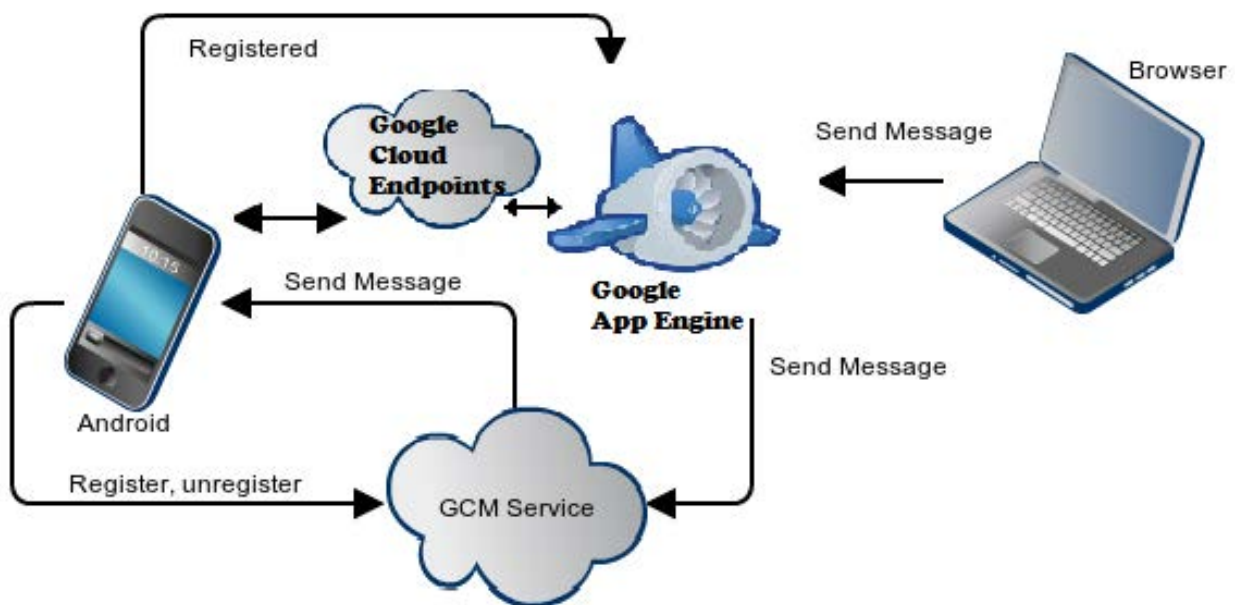


Imagen 8. Arquitectura de sistemas.

En la imagen anterior se observa la arquitectura que siguen las aplicaciones desarrolladas. La aplicación móvil como se ha comentado anteriormente, se encuentra en el dispositivo Android mientras que la web se encuentra alojada en Google App Engine. Para acceder a la aplicación web es necesario disponer de un navegador web y así poder interactuar.

La imagen anterior, además, muestra el flujo de mensajes y las diferentes direcciones que puede llevar. Para que un cliente reciba mensajes del servicio GCM debe haberse registrado anteriormente. Lo mismo ocurre para que reciba mensajes desde el servidor web a través de GCE.

4.5. PATRÓN DE DISEÑO

Para el desarrollo, tanto de la aplicación web como móvil, se ha seguido el patrón de diseño modelo-vista-controlador (MVC) que permite dividir la aplicación en tres módulos independientes, consiguiendo una aplicación muy fácil de mantener y escalar. Los tres componentes que forman dicha arquitectura se caracterizan por lo siguiente:

- Modelo: Se trata del módulo que contiene el dominio de la aplicación, es decir, las clases que contienen los datos necesarios para el funcionamiento de la aplicación. Define el estado actual de la aplicación.
- Vista: Es la representación visual del modelo con la que el usuario es capaz de ver e interactuar. A través de este módulo se accede al modelo para consultarlo, pero nunca para producir cambios en él.
- Controlador: Se trata del módulo que reacciona a las diferentes peticiones de los clientes, ejecutando la acción correspondiente y provocando cambios en el modelo.

Gracias a este tipo de patrón, se puede contar con diferentes equipos especializados de desarrollo, para que cada uno se ocupe de un módulo de la aplicación.

A continuación se detallan, tanto en la aplicación web como móvil, la composición de cada uno de estos módulos, ya que los componentes de ambos sistemas son diferentes.

4.5.1. APLICACIÓN WEB

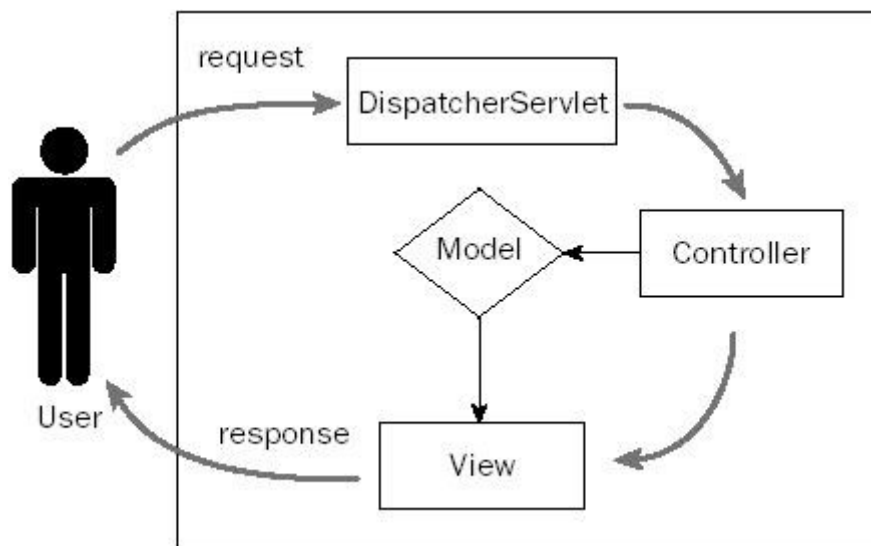


Imagen 9. Arquitectura aplicación web (17).

En la aplicación web, la vista está desarrollada mediante ficheros *JavaServer Pages* (JSP), que contienen a su vez código HTML para realizar la maqueta de la vista, y ficheros CSS que definen los estilos que se muestran al usuario. En el caso de la aplicación desarrollada se ha hecho uso también del *framework* de JavaScript, Bootstrap, el cual cuenta con una serie de etiquetas que se añaden al código HTML para generar la vista.

El modelo, en este caso, está presente en la aplicación desarrollada por medio del datastore proporcionado por Google con el servicio de Google App Engine. Este almacén de datos guarda los datos necesarios para que la aplicación funcione

correctamente. Almacena, por ejemplo la información de los usuarios registrados o del administrador.

El controlador está formado por ficheros Java conocidos como *servlets*, los cuales reciben peticiones y proporcionan la respuesta adecuada.

4.5.2. APLICACIÓN MÓVIL

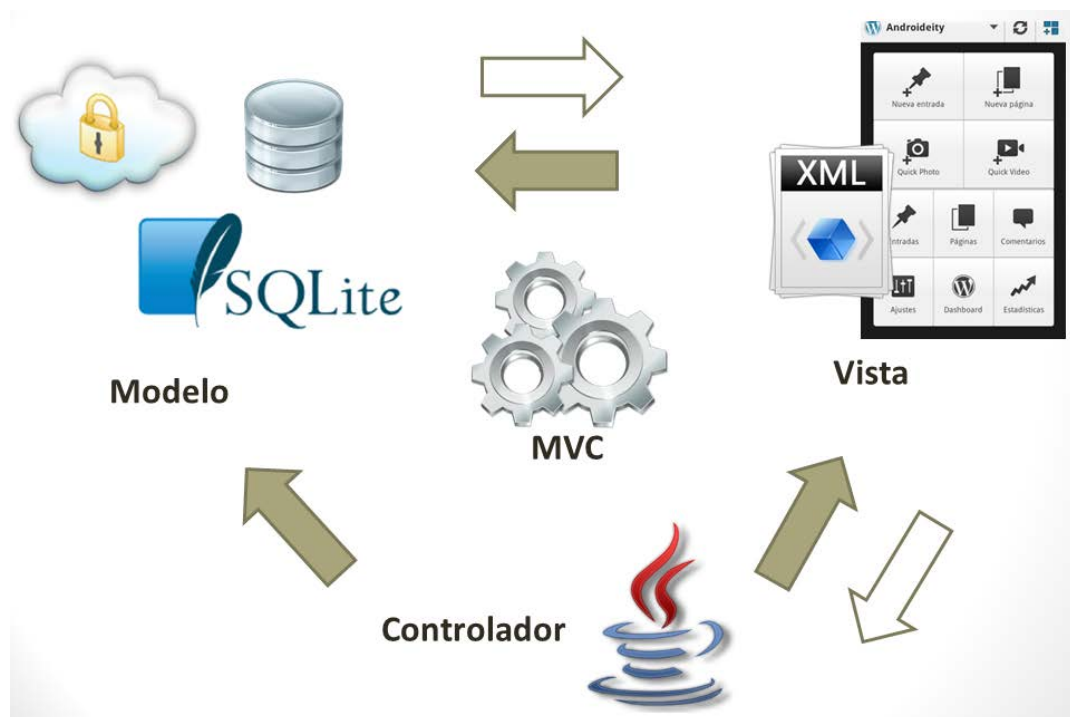


Imagen 10. Arquitectura aplicación Android (18).

En el caso de la aplicación móvil desarrollada para Android los componentes son los mismos pero se desarrollan con diferentes tecnologías.

La vista, en el caso de aplicaciones Android está compuesta por diferentes ficheros XML que expresan el esqueleto básico que tendrá la aplicación, así como los diferentes estilos que se aplicarán a cada una de las vistas generadas en el programa. Son el equivalente a los ficheros HTML y CSS en las aplicaciones web. El modelo está representado por los datos que se encuentran almacenados en la aplicación; en este

caso, la información se guarda en una base de datos relacional SQLite. Por último, el componente controlador está compuesto por diferentes ficheros Java, que se encargan de dar funcionalidad a las diferentes acciones que lleva a cabo el usuario sobre la vista.

4.6. DIAGRAMAS DE ACTIVIDAD

Con los siguientes diagramas se trata de representar los diferentes estados y flujos que se presentan al realizar una acción el usuario, para entender mejor la funcionalidad. En este caso los diagramas de actividad que se van a mostrar son los correspondientes con los casos de uso identificados en la sección 3.4 de la fase de análisis. Los diagramas que se presentan a continuación se componen de:

- Acciones: Se trata de las actividades realizadas en el proceso.
- Condiciones: Situación que debe darse para la elección de un flujo.
- Transiciones: Relación entre las diferentes acciones del proceso.

Para la realización de los diagramas se ha utilizado el estándar UML (Lenguaje Unificado de Modelado). El cual define los siguientes elementos:






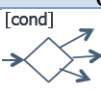


Elemento	Descripción	Representación
Nodo Inicial	Representa el inicio del flujo.	 (19)
Nodo actividad	Representa la actividad en el flujo de ejecución.	 (19)
Flujo de control	Indica la siguiente actividad en el flujo de ejecución.	 (19)
Nodo final de flujo	Indica el final del flujo actual.	 (19)
Nodo final	Indica el fin de la actividad.	 (19)
Nodo de decisión	Marca la existencia de diferentes flujos en función de una condición.	 (19)
Nodo de fusión	Junta diferentes flujos que vienen de un nodo de decisión en uno único.	 (19)
Particiones	Divide el diagrama en diferentes partes dependiendo del sistema responsable.	 (19)

Tabla 57. Elementos diagrama de actividad.

A continuación se presentan los diagramas de actividad pertenecientes a la aplicación web y móvil.

4.6.1. APLICACIÓN WEB

Se presentan los diagramas de actividad pertenecientes a la aplicación web con la cual interactúa el administrador.

4.6.1.1. AUTENTICARSE

Se trata de la actividad en la cual el administrador mediante su nombre de usuario y contraseña accede al sistema. Para que pueda acceder a él, previamente debe estar dado de alta en la base de datos.

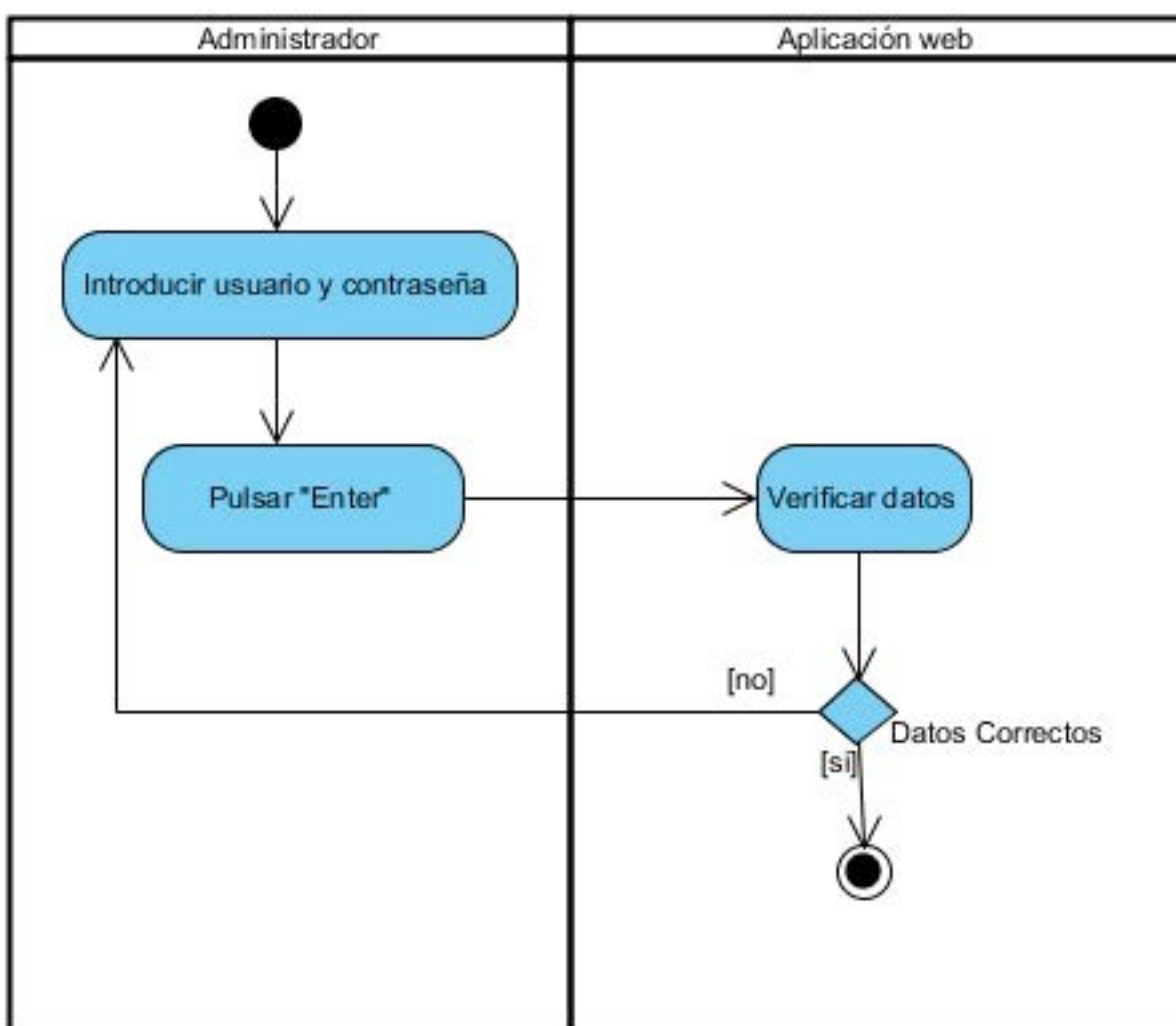


Imagen 11. Diagrama de actividad de autenticación en el sistema.

4.6.1.2. MODIFICAR DATOS DE CLIENTES

Con esta actividad, el administrador desarrolla el proceso de cambiar los parámetros de intervalo de envío de datos y/o el intervalo de captura de datos de los sensores en los dispositivos dados de alta en el sistema. Antes de que se lleve a cabo la acción, el sistema pide una confirmación.

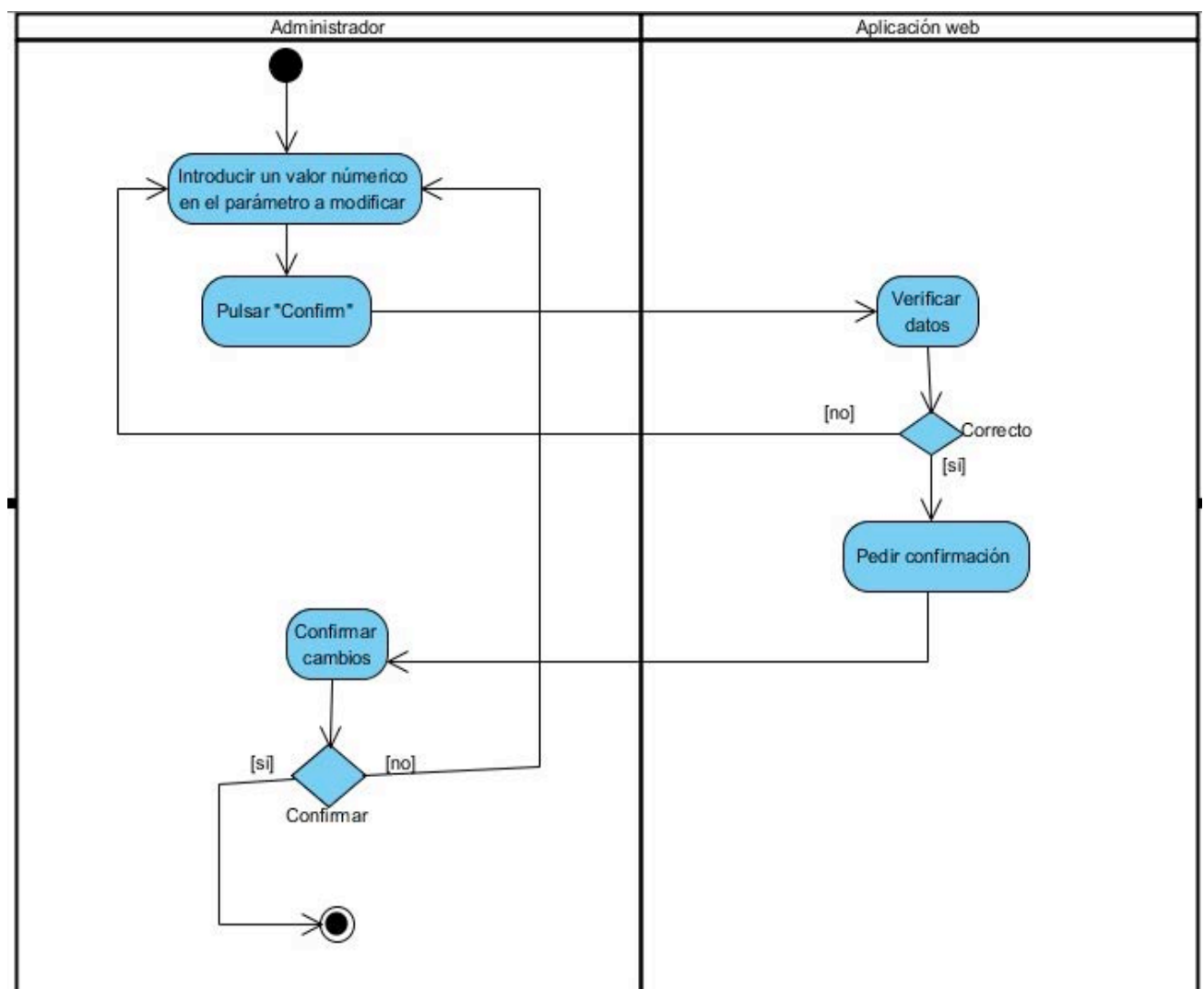


Imagen 12. Diagrama de actividad para modificar parámetros.

4.6.1.3. CONSULTAR DATOS DE CLIENTES

El administrador realiza una consulta en el datastore de la aplicación web a través de un formulario en el que introduce el ID de un usuario dado de alta en el sistema. Si hay coincidencias se muestra una tabla con los datos recogidos de los sensores de dicho usuario, en caso contrario, la tabla no contendrá información. Si se genera la tabla con resultados, el administrador tendrá la opción de exportar los datos a un fichero.

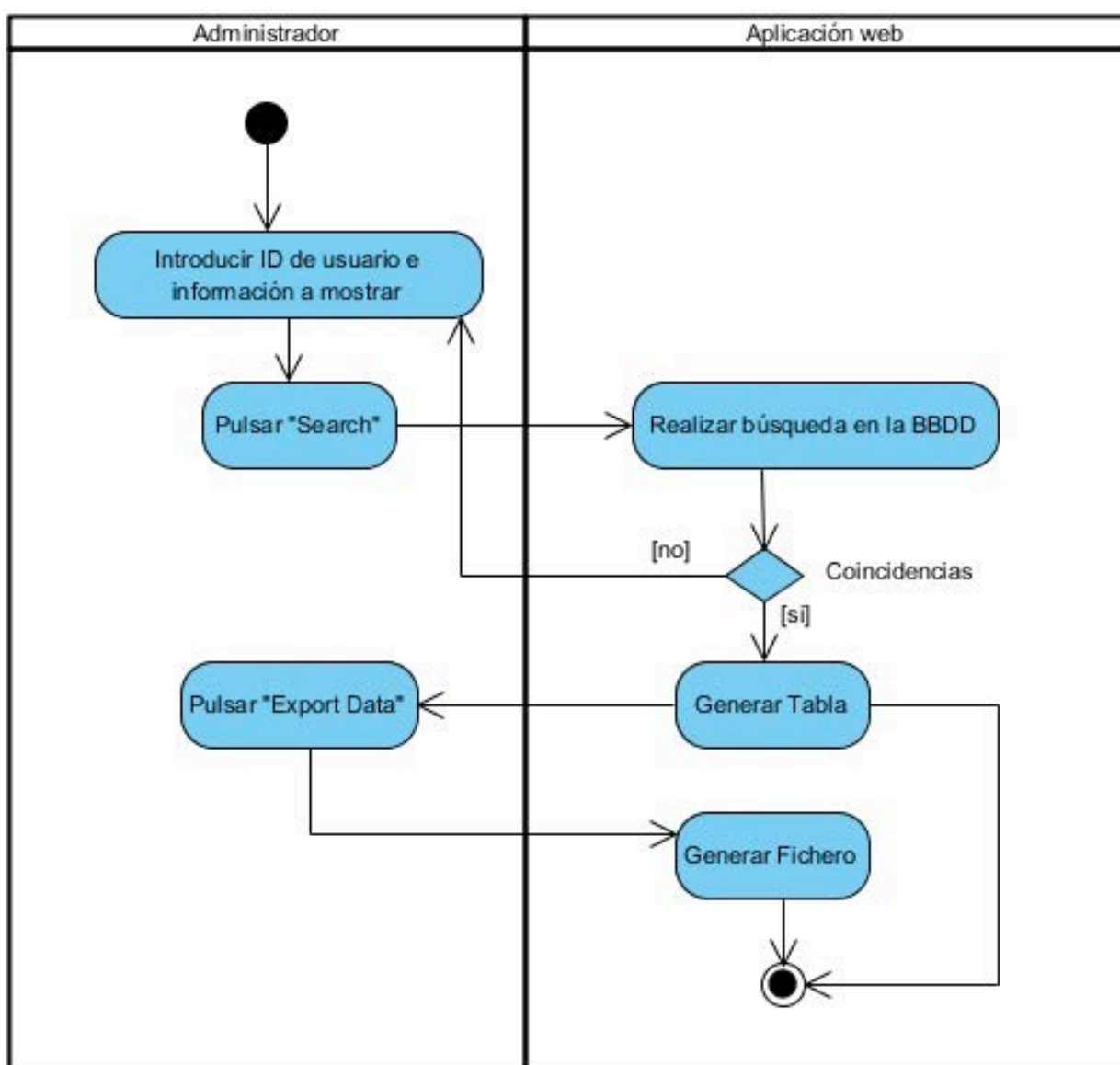


Imagen 13. Diagrama de actividad para la consulta de datos.

4.6.1.4. CERRAR SESIÓN

Con la actividad de cerrar sesión, se le da la posibilidad al administrador de cerrar la sesión activa actualmente y salir del sistema; requiriendo una nueva autenticación en el caso de que se desee acceder de nuevo.

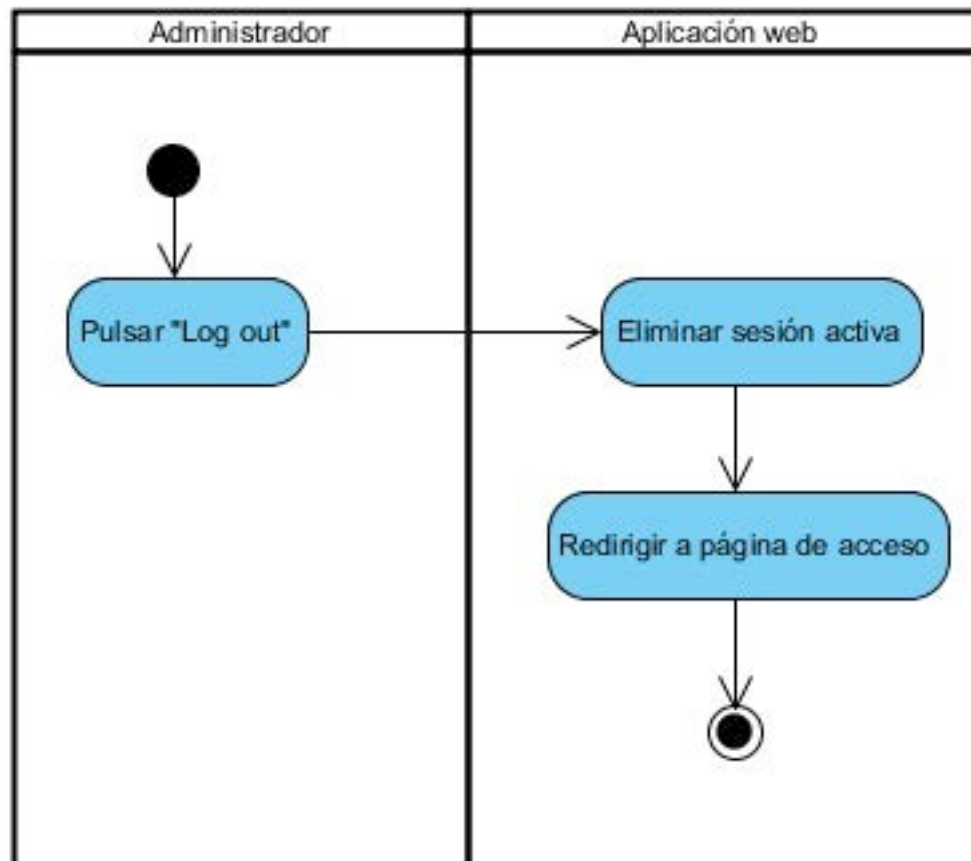


Imagen 14. Diagrama de actividad cierre de sesión.

4.6.2. APLICACIÓN MOVIL

Las actividades que se llevan a cabo en la aplicación móvil son realizadas por el “usuario”. En algunas de ellas se requiere además la acción de la aplicación web. Las actividades identificadas son las siguientes.

4.6.2.1. REGISTRARSE EN EL SISTEMA

El registro en el sistema se realiza únicamente cuando el usuario no está dado de alta en él. Esta acción se lleva a cabo automáticamente la primera vez que se inicia la aplicación, sin la acción del usuario; pero si en algún momento ha decidido darse de baja, para volver a usar la aplicación, deberá realizar esta actividad.

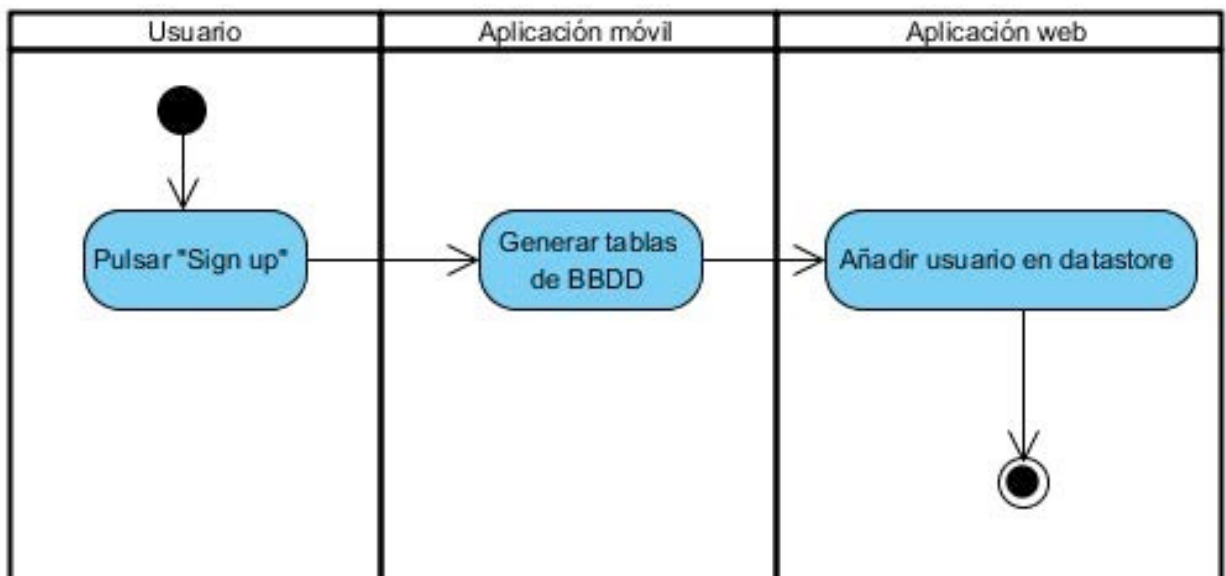


Imagen 15. Diagrama de actividad para registrarse en el sistema.

4.6.2.2. DARSE DE BAJA DEL SISTEMA

Con esta actividad el usuario que se encuentra dado de alta en el sistema se puede dar de baja, eliminándose de la base de datos de la aplicación web y borrando las tablas de la base de datos de la aplicación móvil y, opcionalmente, eliminar los datos recogidos por sus sensores de la base de datos de la aplicación web. Para realizar todas las acciones el sistema pide confirmación por parte del usuario.

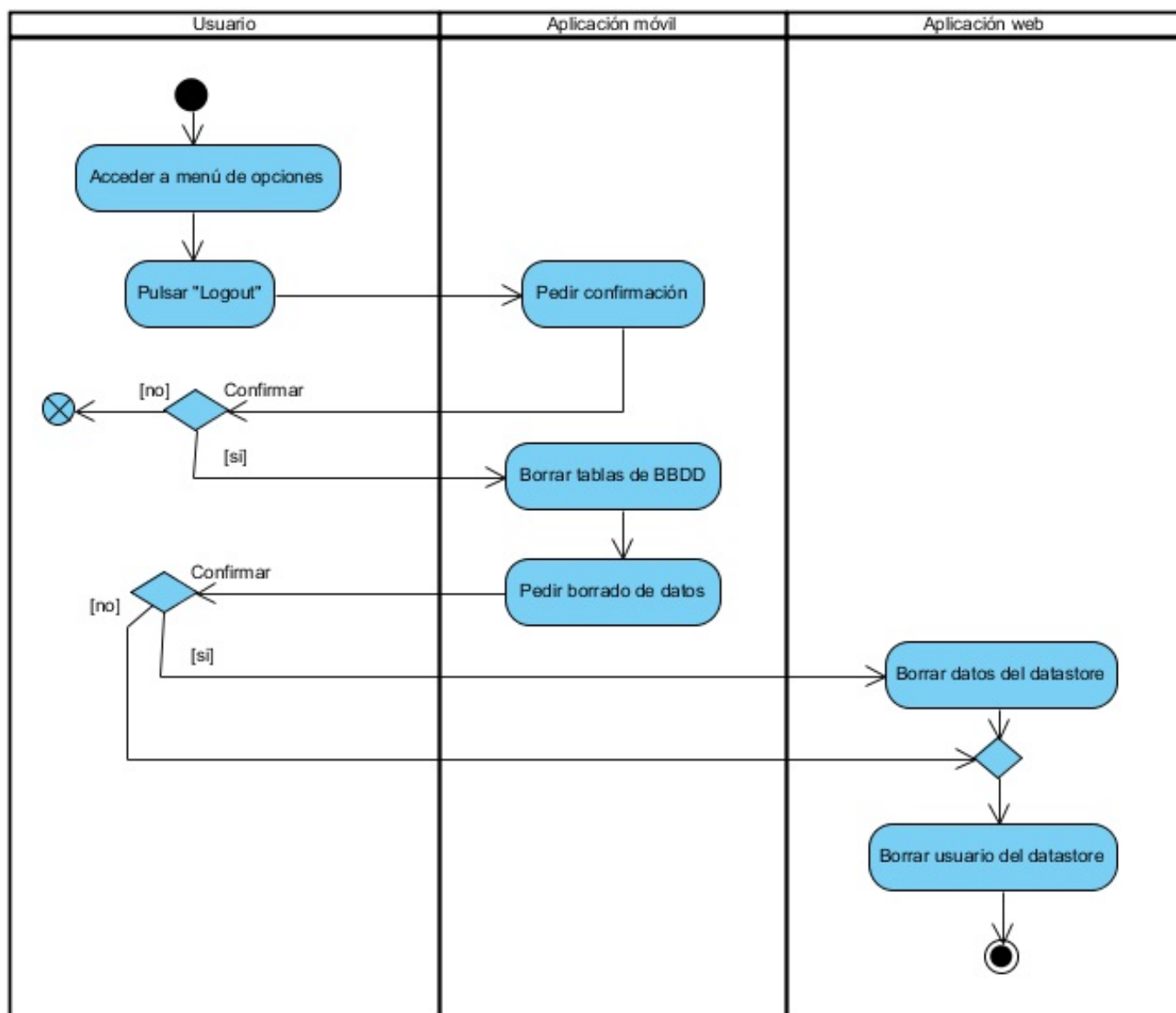


Imagen 16. Diagrama de actividad para darse de baja.

4.6.2.3. CAMBIAR MODO DE EJECUCIÓN

El usuario tiene la opción de cambiar el modo de ejecución de la aplicación en el momento que desee, para ello solo debe pulsar el botón del modo que desea ejecutar; cuando esto sucede, la aplicación lleva a cabo una serie de acciones antes de funcionar en el nuevo modo.

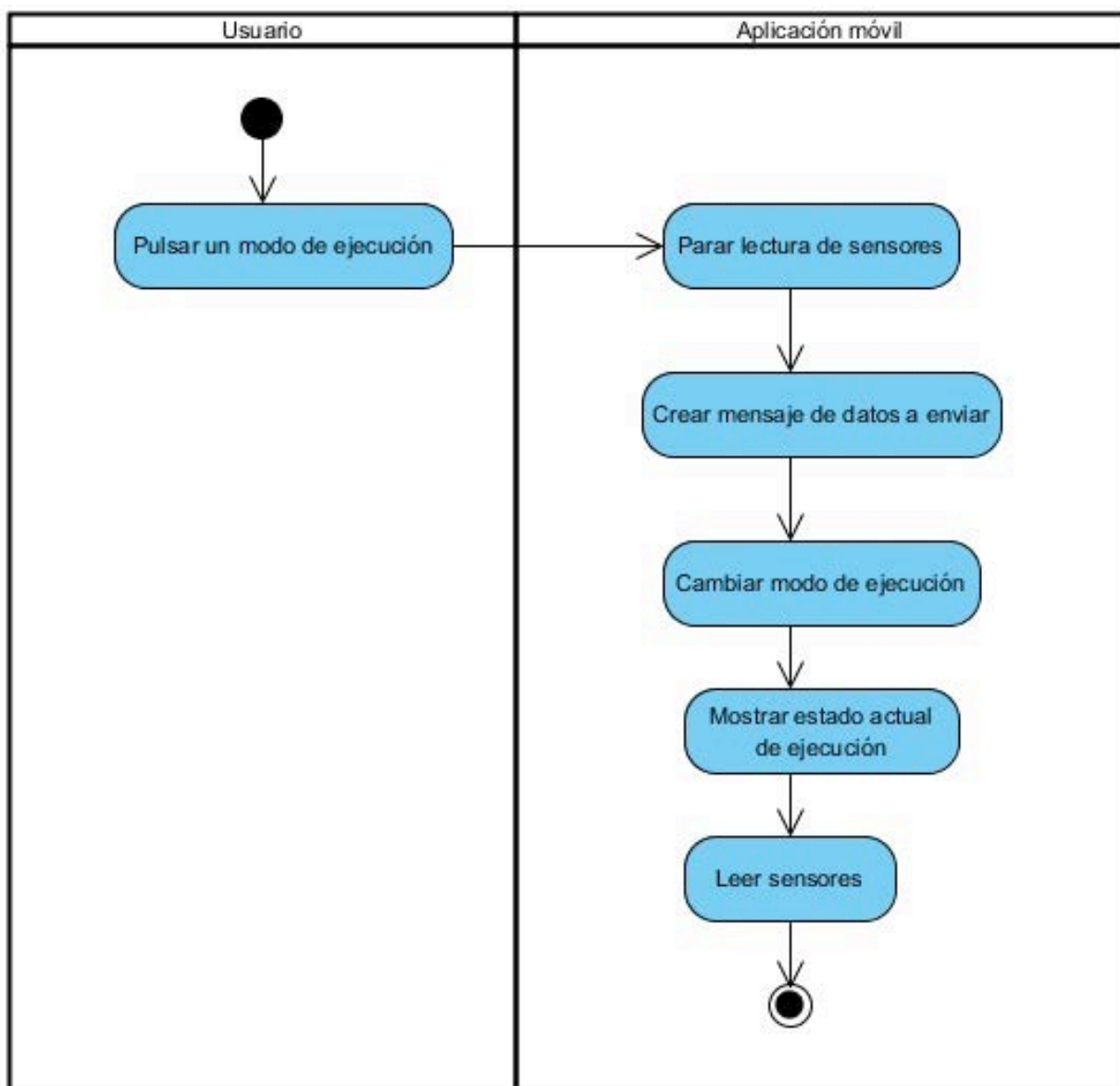


Imagen 17. Diagrama de actividad para cambiar modo de ejecución.

4.6.2.4. PARAR LECTURA DE SENSORES

Si el usuario desea parar la lectura de los sensores y que la aplicación deje de ejecutarse, debe llevar a cabo la presente actividad.

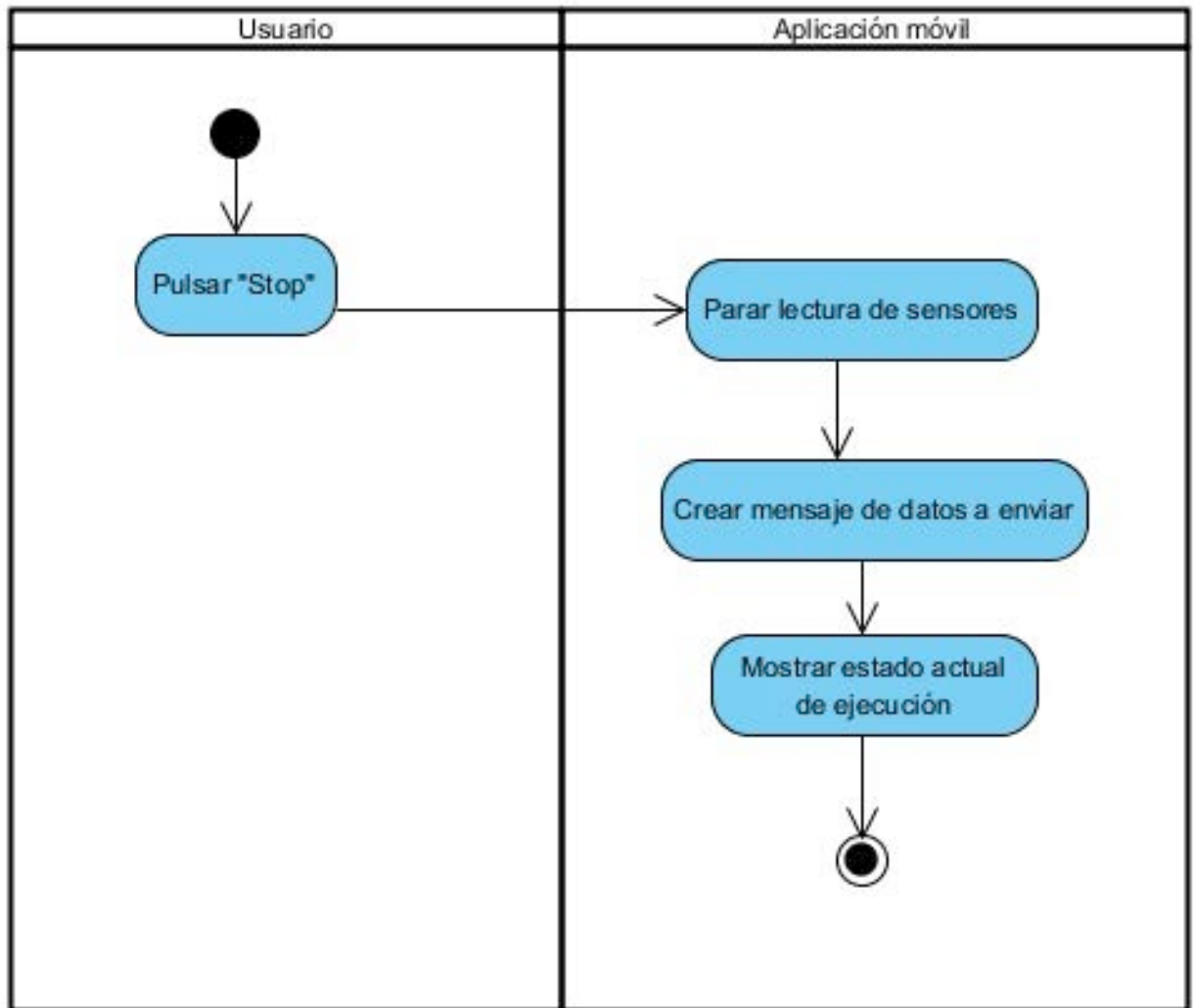


Imagen 18. Diagrama de actividad para parar la lectura de sensores.

4.7. DIAGRAMA DE COMPONENTES

Los diagramas de componentes tratan de dar una visión de la organización y las dependencias que tienen cada uno de los componentes de una aplicación. Cada uno de estos componentes puede contener una o más clases. Además, pueden representar ficheros, bibliotecas compartidas, ejecutables, paquetes o tablas de bases de datos. Gracias a estos diagramas es más fácil analizar qué componentes se pueden reutilizar para futuros sistemas (20).

El sistema desarrollado consta de dos grandes componentes: la aplicación móvil y la web. Estos grandes componentes se encuentran conectados por dos posibles componentes, *Google Cloud Endpoints* y *Google Cloud Messaging* haciendo uso de uno u otro dependiendo de la acción que se vaya a llevar a cabo.

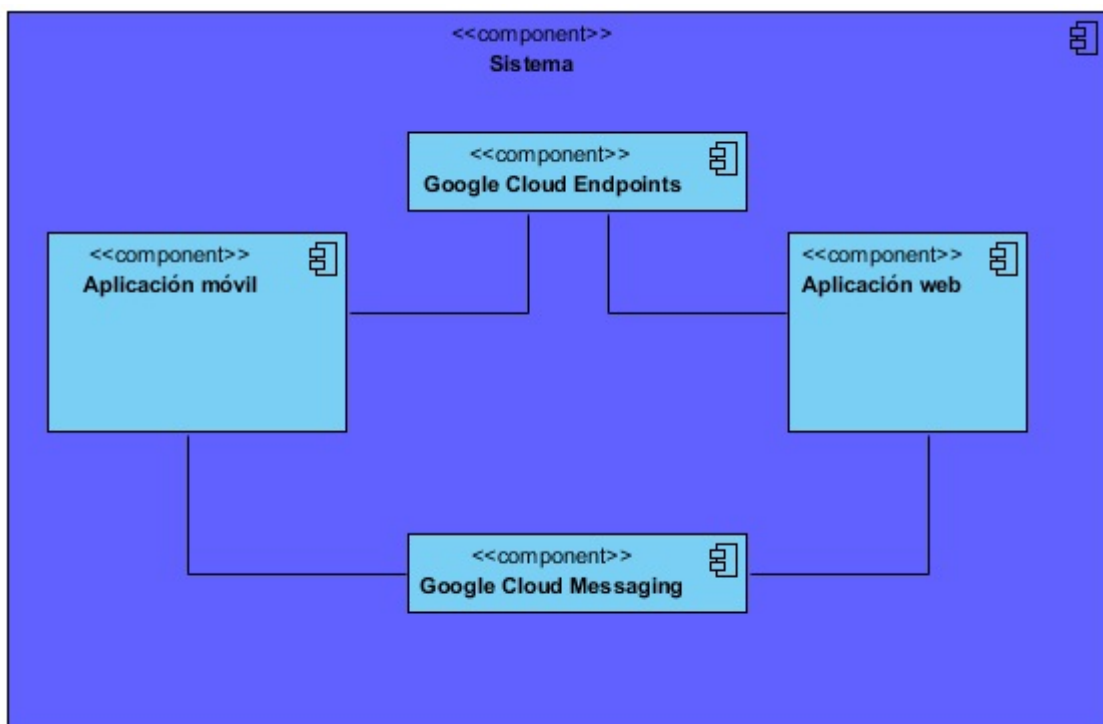


Imagen 19. Diagrama de componentes del sistema.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

A continuación se muestra el diagrama de cada uno de los grandes componentes del sistema desarrollado: la aplicación móvil y la web. Ambos componentes se encuentran conectados según la arquitectura especificada en la sección 4.4.

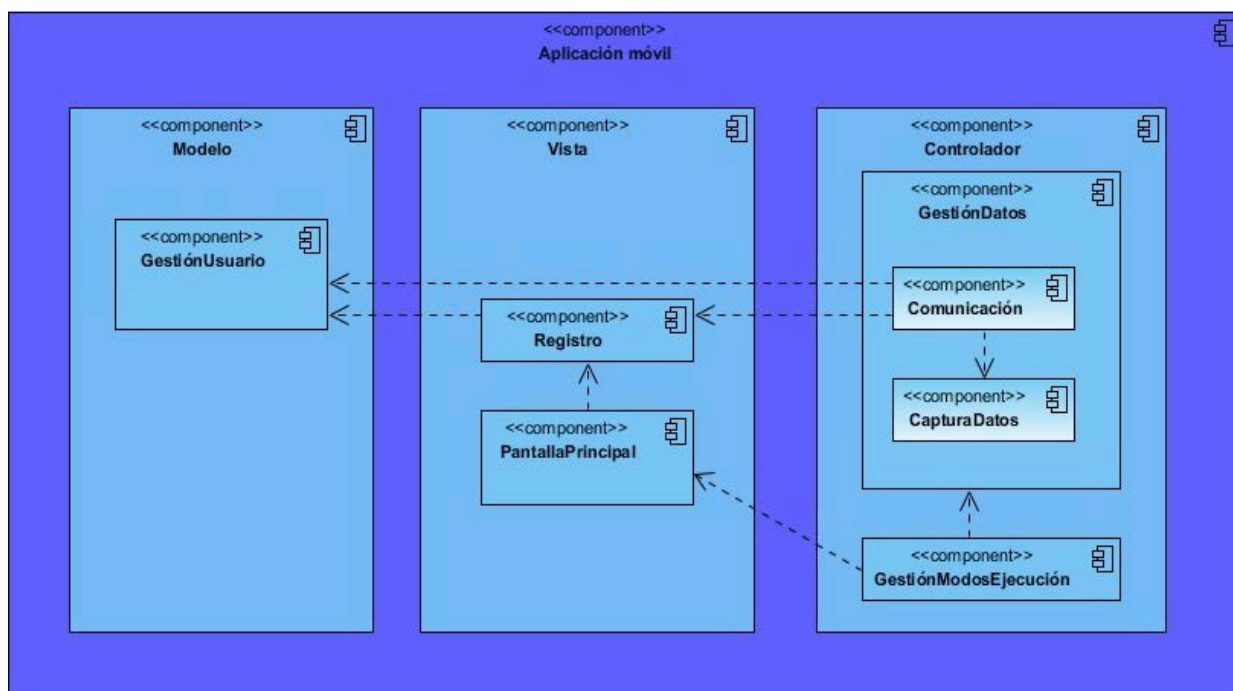


Imagen 20. Diagrama de componentes aplicación móvil.

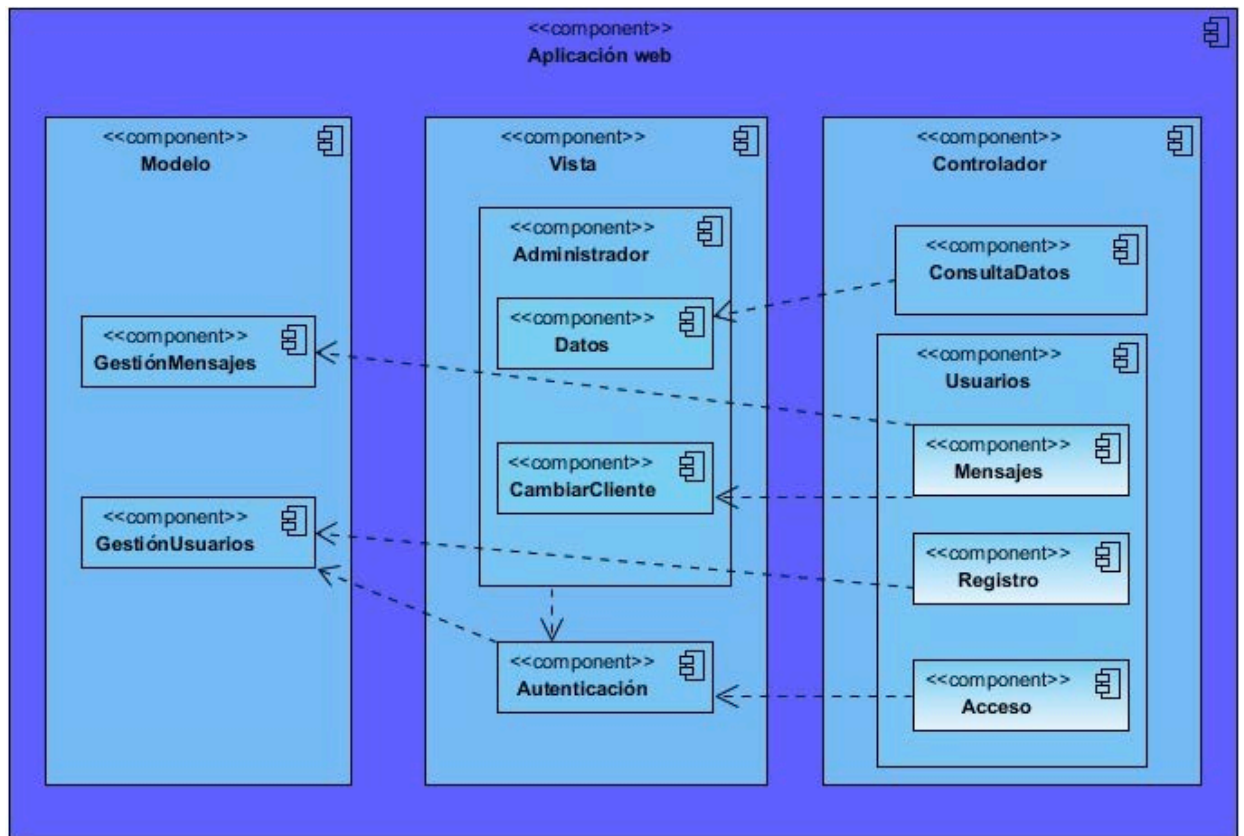
APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

Imagen 21. Diagrama de componentes aplicación web.

4.8. DIAGRAMA DE CLASES

Los diagramas de clases representan las entidades básicas que componen los sistemas, así como sus propiedades y las relaciones que existen entre ellas, entendiendo mejor la información que gestiona el sistema. A continuación se presentan los diagramas correspondientes a la aplicación móvil y web respectivamente.

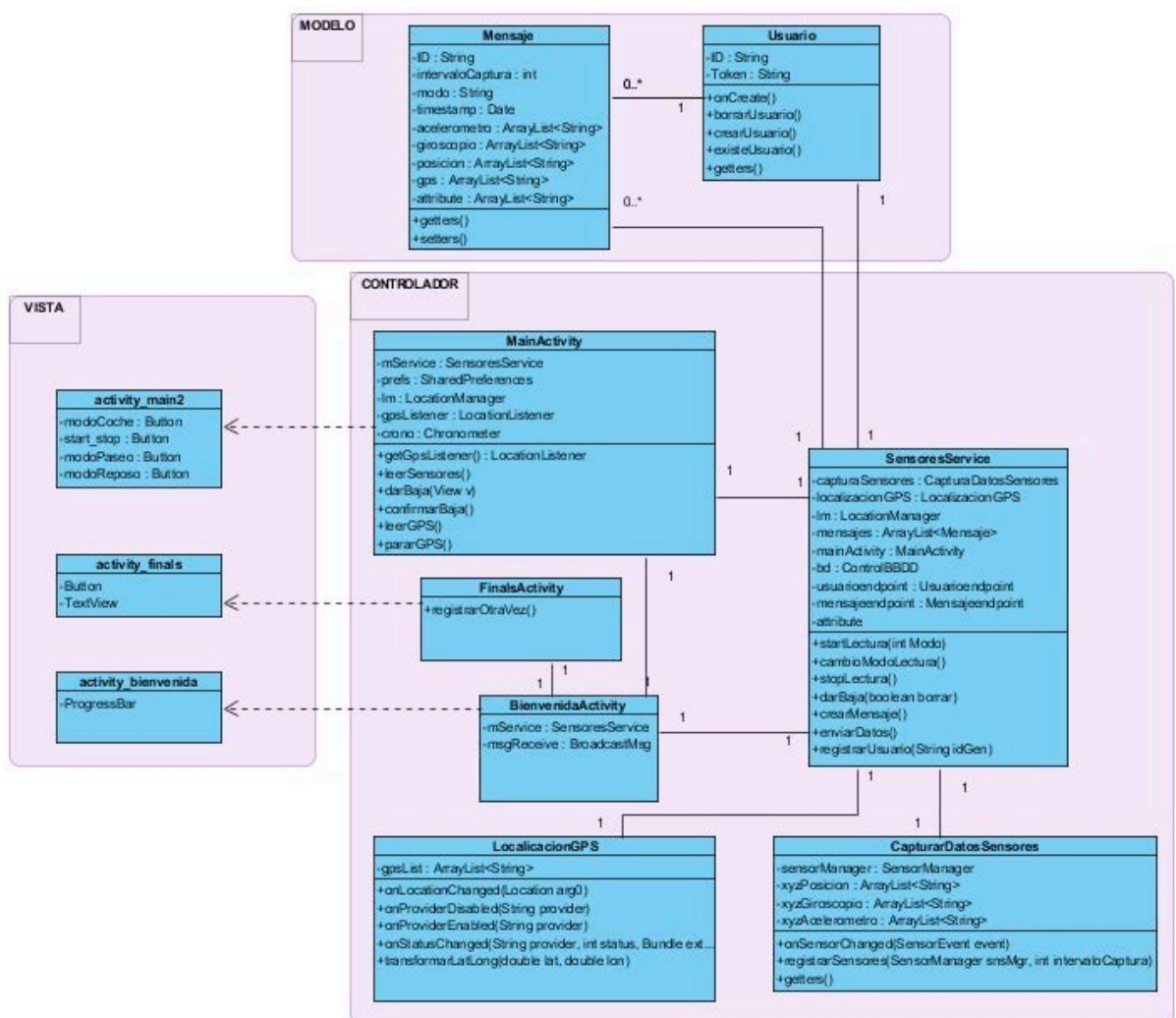


Imagen 22. Diagrama de clases aplicación móvil.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN TELÉFONOS INTELIGENTES

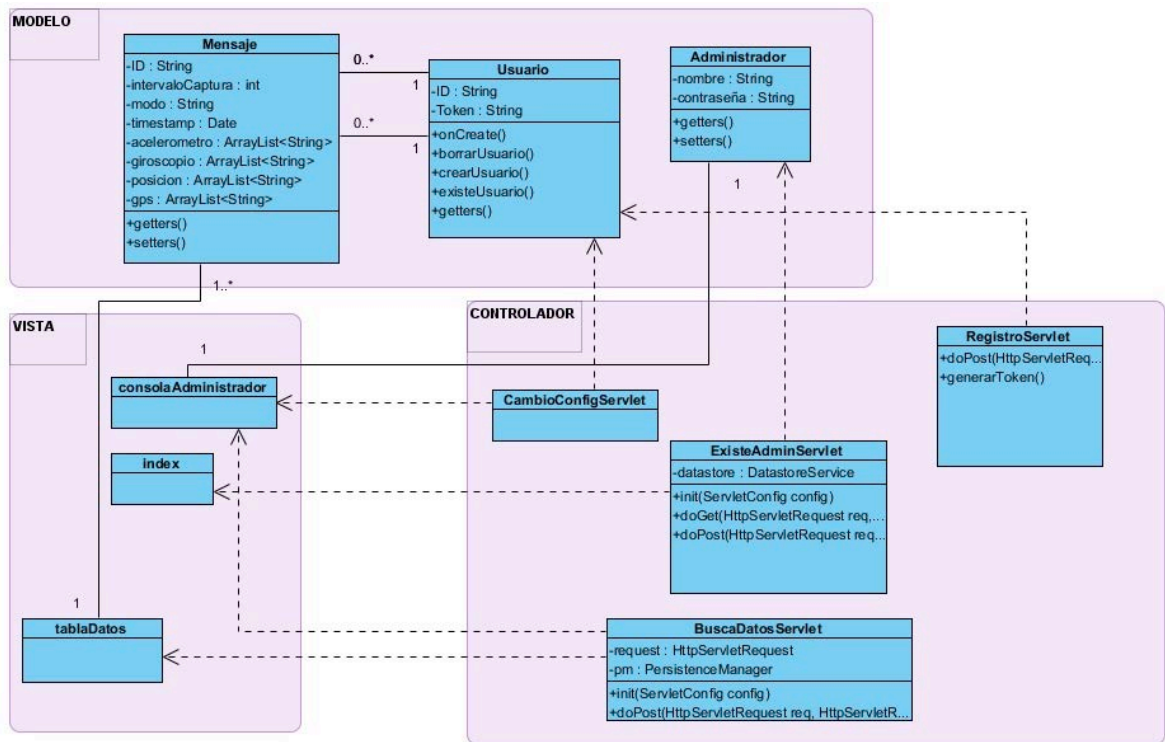


Imagen 23. Diagrama de clases aplicación web.

Como se puede observar en la Imagen 22, el servicio principal de la aplicación llamado *SensoresService*, hace uso de dos clases importantes como son *CapturaDatosSensores* y *LocalizacionGPS* que se encargan de obtener los datos de los sensores y GPS respectivamente, además, coopera con la actividad principal *MainActivity* para recoger los eventos producidos al pulsar los botones de la interfaz el usuario. El uso de líneas discontinuas en las relaciones existentes entre el controlador y la vista significan que hacen uso de esa clase; por ejemplo: la clase *BienvenidaActivity* hace uso de la vista *activity_bienvenida*.

Lo más destacado del diagrama de clases de la aplicación web, es la división de la vista en diferentes clases para hacer un código más fácil de mantener al igual que la división del controlador en diferentes servlets según su funcionalidad.

4.9. INTERFACES

Las interfaces diseñadas se caracterizan por la sencillez y facilidad de uso. En la aplicación móvil están desarrolladas a través de ficheros XML, en los que se usan diferentes etiquetas con las propiedades que se desean para cada elemento de la pantalla a mostrar. En el caso de la aplicación web están desarrolladas con ficheros JSP que hacen uso además del *framework* Bootstrap para que éstas sean más vistosas. Debido también a la sencillez que presenta, se ha optado por no incluir migas de pan ya que el usuario en todo momento sabe dónde se encuentra y cómo ha llegado a la sección en la que está, gracias al menú lateral izquierdo que se facilita.

4.9.1. APLICACIÓN MÓVIL

La aplicación móvil consta únicamente de dos pantallas pero una de ella solo se muestra una vez el usuario se ha dado de baja en el sistema para que tenga la opción de volver a registrarse.



Imagen 24. Pantalla principal.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

- 1.- Pinchando en el menú de ajustes. Da la posibilidad de darse de baja.
- 2.- Botones de cambio de modo.
- 3.- Muestra el tiempo de ejecución de la aplicación en un modo diferente a “inactive”.
- 4.- Permite parar la lectura de los sensores entrando en modo “inactive”.
- 5.- Información del modo actual de ejecución.

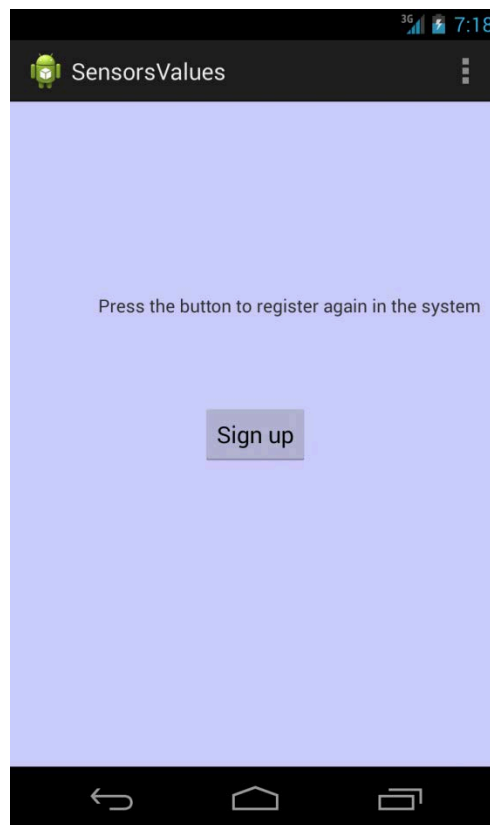
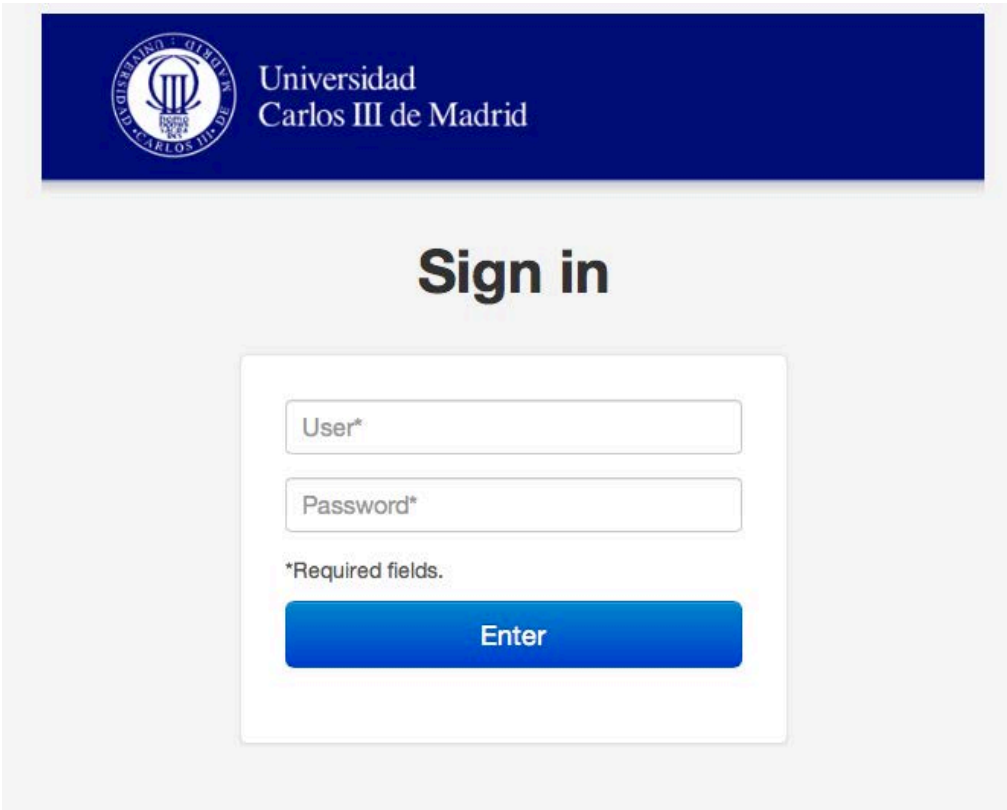


Imagen 25. Pantalla para volver a registrarse.

La pantalla anterior es la que se muestra únicamente cuando un usuario se ha dado de baja en el sistema, para así darle la opción de volver a ingresar en él.

4.9.2. APLICACIÓN WEB

La aplicación web se compone de tres pantallas principales: la de acceso, la de consulta de datos almacenados en el datastore y la pantalla para modificar los parámetros de captura y envío de datos de los sensores al servidor.



Universidad Carlos III de Madrid

Sign in

User*

Password*

*Required fields.

Enter

Imagen 26. Pantalla de acceso a la aplicación web.

A través de esa primera pantalla, el administrador deberá introducir un nombre de usuario y contraseña correctos y a continuación pulsar “Enter”. En el caso de que los datos introducidos sean correctos, se accederá a la pantalla de administración.

UNIVERSIDAD CARLOS III

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES



Imagen 27. Pantalla administración.

Una vez se encuentre en la pantalla de administración, se pueden llevar a cabo dos acciones, consultar datos y ajustes en los clientes que tienen instalada la aplicación móvil y se encuentran dados de alta en el servidor.

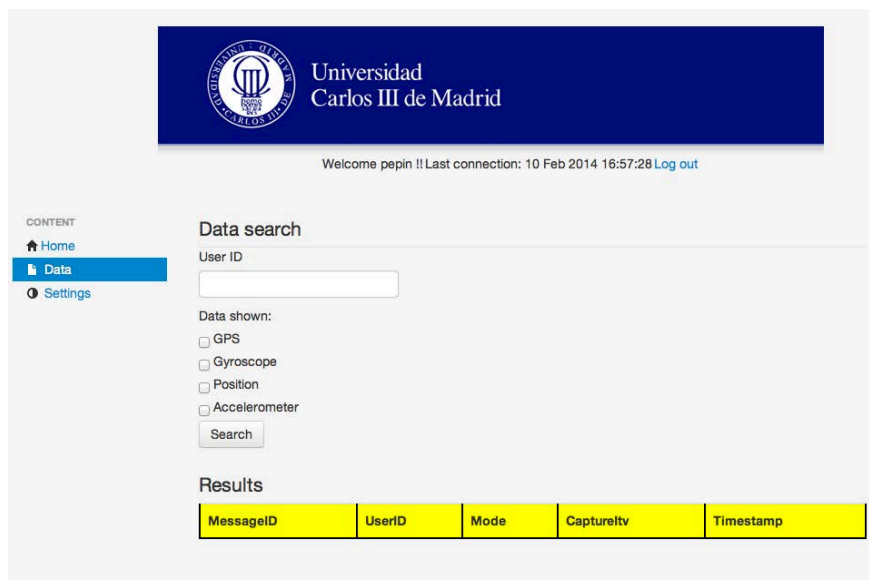


Imagen 28. Pantalla para consultar datos.

UNIVERSIDAD CARLOS III

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

En la pantalla de consulta de datos, cuando se realiza la búsqueda por el id de un usuario que se encuentre en la base de datos, se muestran los resultados en la zona “Results”.

Results				
MessageID	UserID	Mode	CaptureItv	Timestamp
Mensaje(5258163104251904)	5801253128896512	GENERAL	100	Sat Jan 11 20:53:13 CET 2014 20 53 13
Mensaje(5295598039203840)	5801253128896512	GENERAL	100	Sat Jan 11 20:03:22 CET 2014 20 3 22
Mensaje(5304651830263808)	5801253128896512	PASEO	100	Sat Jan 11 20:53:22 CET 2014 20 53 22
Mensaje(5840251431944192)	5801253128896512	GENERAL	100	Sat Jan 11 20:39:49 CET 2014 20 39 49
Mensaje(5846255796224000)	5801253128896512	GENERAL	100	Sun Jan 12 10:57:42 CET 2014 10 57 42

[Export Data](#)

Imagen 29. Resultados consulta.

La última pantalla contiene el formulario para modificar los parámetros de los clientes, y se accede a ella a través de la opción “Settings” de la barra de contenidos.



Universidad
Carlos III de Madrid

Welcome pepin !! Last connection: 11 Feb 2014 17:30:52 [Log out](#)

CONTENT

- Home
- Data
- Settings**

Configuration

Capture interval (ms)

Send Interval (hours)

Confirm

Imagen 30. Pantalla modificación de clientes.

Una vez introducidos los datos únicamente hay que pulsar el botón “Confirm”.

5. IMPLEMENTACIÓN Y PRUEBAS

Una vez realizado el diseño, y en base a él, se comienza la implementación del sistema teniendo también en cuenta los requisitos identificados en el análisis.

En el presente capítulo se comentaran los aspectos más importantes y relevantes a la hora de codificar la aplicación móvil y web, así como las decisiones tomadas en cada uno de estos aspectos. Por último se presentarán los resultados obtenidos en la realización de las pruebas especificadas en la sección 3.10.

La aplicación web se encuentra disponible en la dirección <https://evaluasensores.appspot.com> mientras que la móvil puede descargarse en <https://dl.dropboxusercontent.com/u/26021357/ProyectoSensores.apk>

5.4. CONCURRENCIA

Para que la aplicación móvil no se quede mucho tiempo realizando tareas pesadas provocando el bloqueo de la interfaz del usuario, se hace uso de diferentes hilos. En este caso tareas como el envío de datos al servidor, la creación de mensajes a enviar, el registro de un usuario y el borrado del mismo, son llevadas a cabo por diferentes hilos dentro del servicio principal de la aplicación, *SensoresService.java*.

Un problema de concurrencia es aquel en el que dos hilos o procesos diferentes, acceden a una misma zona de memoria, y al menos uno de ellos intenta realizar una modificación. En nuestro caso hay dos hilos que presentan conflictos al acceder a una misma variable. Se trata del hilo encargado de crear un mensaje y añadirlo a la lista de mensajes pendientes de enviar, y del hilo encargado de enviar los mensajes, que para ello, debe recorrer dicha lista.

Además de ese problema, los mensajes pendientes de enviar están compuestos de listas de valores de sensores que añaden los diferentes *listeners* de los sensores, por lo que no se puede garantizar que en el momento de añadir un mensaje

a la lista de mensajes pendientes, ese mensaje no tenga aún una lista a la que le faltan valores por introducir provocando un problema de concurrencia.

Para solucionar el problema se ha decidido que, a la hora de enviar los mensajes, previamente, el hilo encargado de la tarea deja de leer el valor de los sensores y el GPS, en el caso de que se encuentre activado; se “duerme” 500 milisegundos para dar tiempo a que todas las listas se encuentren completas, crea el mensaje en caso de que exista y procede al envío de mensajes pendientes. Una vez realizado el envío, vuelve a registrar los sensores para que continúen recogiendo los valores leídos.

Las líneas de código más significativas son las siguientes.

```
//Paramos la lectura de los sensores para evitar problemas de concurrencia.  
sensorManager.unregisterListener(capturaSensores);  
MainActivity.pararGPS();  
  
//Esperamos a procesar todos los datos  
Thread.sleep(500);
```

Imagen 31. Parar lectura de sensores, GPS y dormirse.

Indicar que el tiempo de 500 milisegundos se ha elegido ya que funciona en las pruebas realizadas con dos dispositivos interactuando con la aplicación web, pero al no haberlo probado con una gran cantidad de usuarios, no se puede saber si funcionará o no.

5.5. COMUNICACIÓN CON APLICACIÓN MÓVIL

Para que la aplicación web realice el intercambio de datos con la aplicación móvil se ha hecho uso del servicio *Google Cloud Messaging*. Los parámetros que la aplicación web puede mandar a la móvil son los tiempos de envío de mensajes al servidor y el intervalo de captura de datos de los sensores de los clientes. Para ello, la aplicación web hace uso de esta tecnología permitiendo de una forma muy simple

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN TELÉFONOS INTELIGENTES

enviar la misma información a todos los clientes que se encuentran dados de alta en el servicio. Para ello, en el momento del registro del usuario en la aplicación móvil, se le asigna un identificador para que la aplicación web pueda comunicarse con él a través de *Google Cloud Messaging*. Un ejemplo de ID sería el siguiente:

Usuario Entities

Prev 20 1-3 Next 20	
<input type="checkbox"/>	ID/Name idGCM
<input type="checkbox"/>	id=5815383571300352 APA91bEvPmfp_Jpzxm68KuJbbQZGBVCEnp3N69zYrce-gCJ_0DGOjz0jrNQ8dJKYB-2qGdQsWah6VunO8HR_IgILgyT64SGsVHH4rzOm3QblgTPwXha6Bd7cQ1bCOqkxU7zfTguMlbql1VLF_HQY3-1hVxv8h4Sso8OwhMAX9fG-A6Lq2RofU

Imagen 32. Ejemplo de identificador de Google Cloud Messaging.

El envío de datos se realiza desde un *servlet* que recoge los datos del siguiente formulario:

Imagen 33. Formulario para modificar parámetros de clientes.

El *servlet*, una vez recoge los datos a enviar, comprueba si únicamente hay un cliente almacenado en la base de datos o hay varios, ya que el código desarrollado

para realizar el envío es diferente en un caso y en otro. El código correspondiente a esta acción es el siguiente:

```
Sender sender = new Sender(API_KEY);  
Message.Builder msgBuilder = new Message.Builder();  
  
if(captura !=null && !captura.equals("")){  
    msgBuilder.addData("captura", captura);  
}  
if(envio != null && !envio.equals("")){  
    msgBuilder.addData("envio", envio);  
}
```

Imagen 34. Envío de datos usando Google Cloud Messaging I.

En esa primera parte, se crea un *Sender* que es el encargado finalmente de realizar el envío de los datos a través de su método *send()*. Es aquí donde se añaden los datos a enviar a los clientes por medio de *Message.Builder* y su método *addData()*. A continuación se muestra el código en el que finalmente se envía el mensaje o mensajes.

```
List<String> regIds = new ArrayList<String>();

for(int i = 0; i<users.size(); ++i){
    regIds.add(users.get(i).getIdGCM());
}

List<Result> results;
if (users.size() == 1) {
    // Envío de un unico mensaje
    String registrationId = users.get(0).getIdGCM();
    Result result = sender.send(msgBuilder.build(), registrationId, 5);
    results = Arrays.asList(result);
    request.setAttribute("envioCorrecto", "true");
} else {
    // Se envia un multicast usando JSON.
    MulticastResult result = sender.send(msgBuilder.build(), regIds, 5);
    results = result.getResults();
    request.setAttribute("envioCorrecto", "false");
}
```

Imagen 35. Envío de datos usando Google Cloud Messaging II.

Como se puede observar, en función de si hay un cliente registrado o varios, el envío de datos se realiza de manera diferente realizando un *multicast* si existen varios.

5.6. ALMACENAMIENTO DE DATOS

Tanto para la aplicación móvil como para la web, se requiere el almacenamiento de algunos datos para el correcto funcionamiento de las aplicaciones. En ambos sistemas hay varias alternativas.

Para la aplicación móvil se han usado dos formas de almacenamiento de datos: datos almacenados en el *SharedPreferences* y en la base de datos SQLite de la aplicación. En el *SharedPreferences* se ha optado por almacenar información acerca de la configuración del sistema como son los tiempos de captura de los sensores o los tiempos de envío mientras que en la base de datos SQLite se almacena la información del usuario que posee la aplicación con su identificador.

En la aplicación web se ha optado por almacenar los datos en un almacén de datos en el que se guardan objetos. No se trata de una base de datos relacional. Al tratarse de objetos, está enfocada al paradigma de la programación orientada a objetos, y al realizar una consulta los obtenemos con todas sus propiedades, lo que resulta muy cómodo en este tipo de aplicaciones.

Para poder almacenar objetos de tipo “Usuario” y “Mensaje” en el datastore desde la aplicación web, se han implementado las clases de JDO (Java Data Object) con los atributos de cada uno de los tipos de objeto así como sus diferentes *setters* y *getters* para dar valor a los atributos u obtenerlos respectivamente.

Para que desde la aplicación móvil se puedan crear y enviar a la aplicación web, se hace uso de *Google Cloud Endpoints*. Se trata de un proceso automático una vez desarrolladas las clases de JDO. Por ello, una vez desarrolladas las entidades de la aplicación, el siguiente paso es crear un *endpoint* por cada una de ellas. Cada *endpoint* contiene la descripción de la entidad correspondiente así como los diferentes métodos que puede ejecutar. Por último, solo sería necesario generar las librerías para la aplicación cliente. Tanto la generación del *endpoint* como la generación de las librerías cliente se realiza por medio de Eclipse gracias al *plugin* de Google App Engine.

Una vez generado el *endpoint*, se pueden realizar modificaciones en el código o añadir nuevos métodos.

Inicialmente, al usar los *endpoint* se producía un error al insertar objetos, por lo que hubo que añadir una condición al código comprobando antes de que se verifique si ya existe, si tiene asignado ID o no. El código para la inserción de un usuario queda de esta forma:

```
@ApiMethod(name = "insertUsuario")
public Usuario insertUsuario(Usuario usuario) {
    PersistenceManager mgr = getPersistenceManager();
    try {
        if(usuario.getId() != null){
            if (containsUsuario(usuario)) {
                throw new EntityExistsException("Object already exists");
            }
        }
        mgr.makePersistent(usuario);
    } finally {
        mgr.close();
    }
    return usuario;
}
```

Imagen 36. Código endpoint usuario.

Como se puede observar en la línea resaltada, antes de ver si el usuario existe ya en el datastore, se comprueba si tiene un ID asignado; de no tenerlo, significa que aún no se encuentra y por lo tanto hay que añadirlo en el datastore el cual, le asignará uno automáticamente.

5.7. PRIVACIDAD DEL USUARIO

Al recoger la posición del usuario, a partir del GPS, el sistema almacena su latitud y longitud entre otros valores. Analizando las coordenadas recogidas de cada uno de los usuarios, podríamos llegar a saber información personal como su lugar de residencia.

Con el fin de evitar que ocurra este tipo de situaciones, cuando se recogen los valores captados por el GPS del dispositivo, se les aplica una transformación sumándoles un valor tanto a la latitud como a la longitud y a continuación una rotación.

Para que se trate de unos valores totalmente personales, al iniciarse la aplicación por primera vez, se generan dos números aleatorios entre 0 y 100, y un ángulo de rotación entre 0 y 120. Estos valores son almacenados en el *SharedPreferences* de la aplicación. Realizándolo de esta manera, sería mucho más difícil averiguar el lugar de residencia de un usuario, siempre y cuando teniendo en cuenta la orografía en torno a la que se desplaza el usuario ya que, si por ejemplo, se recogen unas coordenadas en las que se puede dibujar un camino a lo largo de ellas, por mucho que se giren y desplacen esas coordenadas, si únicamente cerca de ese lugar hay una carretera que tenga una forma parecida, se podría intuir que va a través de ella, pero aun así el problema se complica mucho más; además, con esta modificación la aplicación seguiría cumpliendo su función correctamente

```
/**
 * Metodo para generar los numeros para realizar las transformaciones en las coordenadas xyz del sensor de posicion
 * para salvaguardar la privacidad del usuario.
 */
private void generarCoordenadas(){

    int x = (int)(Math.random()*100);
    int y = (int)(Math.random()*100);

    editor.putInt("cambioX", x);
    editor.putInt("cambioY", y);
    editor.putFloat("angulo", generarAngulo());

}

/**
 * Metodo complementario a generarCoordenadas() que genera un angulo de giro.
 * @see generarCoordenadas()
 * @return Angulo de giro para la transformacion
 */
private float generarAngulo(){

    float result;

    result = (float) (Math.random()*120);

    return result;

}
```

Imagen 37. Código para la generación de números.

En la imagen anterior se encuentra el código perteneciente a la generación de los dos números para la modificación de la longitud y latitud así como del ángulo de rotación. La utilización de estos valores se indica en la siguiente imagen observándose como se obtienen del *SharedPreferences*, y en el caso de que no se encuentren dichos valores, que le suma un cero o noventa si se trata del ángulo.

```

/**
 * Metodo que realiza la transformacion de las coordenadas recibidas por el GPS. La transformacion es la siguiente
 *      x' = x.cos(a) - y.sen(a)
 *      y' = x.sen(a) + y.cos(a)
 *      z' = z (coordenada que no varia )
 * @param x1 Coordenada latitud captada por el GPS
 * @param y1 Coordenada Longitud captada por el GPS
 * @return Cadena con la Latitud+Longitud.
 */
private String transformarLatLong(double lat, double lon){

    String conversion;
    double x2 = lat + prefs.getInt("cambioX", 0);
    Log.i("LOCALIZACIONGPS ----> transformarLatLong()", "LA X2 es: "+x2);
    double y2 = lon + prefs.getInt("cambioY", 0);
    Log.i("LOCALIZACIONGPS ----> transformarLatLong()", "LA Y2 es: "+y2);
    double x3;
    double y3;

    float alfa = prefs.getFloat("angulo", 90);

    x3 = x2*Math.cos(alfa) - y2*Math.sin(alfa);
    y3 = x2*Math.sin(alfa) + y2*Math.cos(alfa);

    conversion = "Latitud="+Double.toString(x3) + " Longitud="+ Double.toString(y3)+" ";

    return conversion;
}

```

Imagen 38. Código para modificar latitud y longitud.

5.8. REGISTRO Y LECTURA DE SENSORES

Para poder leer el valor de los sensores del dispositivo, previamente hay que registrarlos para que al producirse cambios en él, se detecten por un *listener*. Para evitar que se produzcan errores, lo primero que se debe realizar es obtener la lista de sensores disponibles en el dispositivo a través de un objeto *SensorManager*. Una vez confirmamos que se encuentra disponible el sensor deseado, se procede al registro del mismo asignándole un *listener*, que no es más que un objeto que se encargará de detectar si se han producido cambios, un intervalo de captura de datos y el propio sensor al que se le asocia el *listener*.

Una vez que se ha realizado el registro de los sensores deseados, ya se pueden captar automáticamente cada “x” tiempo (siendo “x” el intervalo de captura de datos) los valores recogidos por los sensores a través del método *onSensorChanged(SensorEvent event)*.

A continuación se muestra el código de los pasos descritos anteriormente para el sensor de aceleración:

```
public void registrarSensores(SensorManager sensMgr, int intervaloCaptura) {  
  
    sensorManager = sensMgr;  
  
    List<Sensor> listaSensores = sensorManager.getSensorList(Sensor.TYPE_ALL);  
  
    for(Sensor sensor: listaSensores) {  
        Log.i("CapturarDatosSensores__registrarSensores() -----> ListaSensores", "SENSOR "+sensor.getName());  
  
        switch (sensor.getType()) {  
  
            case Sensor.TYPE_ACCELEROMETER:  
                sensorManager.registerListener(this, sensor, intervaloCaptura);  
                xyzAceleracion = new ArrayList<String>();  
  
                break;
```

Imagen 39. Código del registro de un sensor.

```
@Override
public void onSensorChanged(SensorEvent event) {

    synchronized (this) {
        switch(event.sensor.getType()) {

            case Sensor.TYPE_ACCELEROMETER:
                String auxB="";

                for (int i=0 ; i<3 ; i++) {
                    switch (i){

                        case 0:
                            auxB += "x="+Double.toString(event.values[i])+ " ";
                            break;
                        case 1:
                            auxB += "y="+Double.toString(event.values[i])+ " ";
                            break;
                        case 2:
                            auxB += "z="+Double.toString(event.values[i]);
                            break;
                    }
                }
                xyzAceleracion.add(auxB);
                break;
            }
        }
    }
}
```

Imagen 40. Código para leer los valores de un sensor.

En la imagen anterior se observa el código perteneciente a la lectura de los valores recogidos por el sensor de aceleración. Para ello, el método recibe un evento producido, si pertenece al sensor de aceleración, se procede a su lectura. Como dicho sensor tiene tres valores, uno para cada eje del eje de coordenadas xyz, se debe acceder a cada uno de ellos a través del objeto *SensorEvent*.

5.9. RESULTADOS DE LAS PRUEBAS

En este último apartado del capítulo se presenta la tabla especificada en la sección 3.10 con el resultado obtenido en la ejecución de cada una de las pruebas. Todas las pruebas se han superado.

PRUEBA	RESULTADO
PR-01	OK
PR_02	OK
PR-03	OK
PR-04	OK
PR-05	OK
PR-06	OK
PR-07	OK
PR-08	OK
PR-09	OK
PR-10	OK
PR-11	OK
PR-12	OK
PR-13	OK
PR-14	OK
PR-15	OK
PR-16	OK
PR-17	OK
PR-18	OK

Tabla 58. Resultado de las pruebas ejecutadas.

6. CONCLUSIONES Y TRABAJOS FUTUROS

En este Trabajo de Fin de Grado se ha desarrollado una aplicación móvil para el sistema operativo Android y una aplicación web para interactuar con la aplicación móvil y almacenar los datos enviados por ella.

Para esto, primero se ha realizado un análisis de la evolución del uso de sensores en los dispositivos móviles, así como de las diferentes tecnologías necesarias para desarrollar este tipo de aplicaciones, que empresas como Google están desarrollando para facilitar la vida a los desarrolladores de software. Tecnologías, por ejemplo, como Google Cloud Endpoints y Google Cloud Messaging para la comunicación entre ambas. Una vez analizado el entorno actual en el que se van a desarrollar las dos aplicaciones, se ha llevado a cabo un análisis de los diferentes requisitos de software que debían cumplir así como los diferentes casos de uso identificados. Habiendo realizado el análisis de ambos sistemas, se ha especificado un plan de pruebas a ejecutar una vez implementadas aplicaciones.

Antes de realizar la codificación de los sistemas, se ha realizado un diseño de su arquitectura; ambos sistemas se basan en el patrón de diseño MVC (Modelo-Vista-Controlador) para facilitar de ese modo la reutilización de componentes y el mantenimiento de los mismos. En la última fase del desarrollo se ha procedido a la implementación de los sistemas así como a llevar a cabo el plan de pruebas especificado en el análisis.

Todo el sistema desarrollado cumple su función obteniéndose los datos de los sensores de aceleración, giroscopio, posición y GPS a través de las aplicaciones móviles y recibiendo en la aplicación web que los almacena en la base de datos para posteriores consultas. El sistema completo ha sido utilizado con diferentes dispositivos como el Samsung Galaxy S3, Samsung Galaxy Note II, Google Nexus 10 y Google Nexus 5 así como

con diferentes versiones de sistema operativo Android: 4.1.2, 4.3 y 4.4.2. En todos los casos funciona.

6.4. CONCLUSIONES PERSONALES

En el aspecto personal, el trabajo realizado ha supuesto un gran reto al tener que partir de cero en la realización del análisis, diseño e implementación, con la investigación inicial de las tecnologías actuales y más novedosas (como son las mencionadas anteriormente) acarreando muchas horas de documentación y comprensión para poder hacer uso de ellas de la manera correcta y más eficiente posible. Al tratarse de un desarrollo partiendo desde cero, he sido realmente consciente del ciclo completo del desarrollo de software en el que te encuentras continuamente realizando modificaciones, y dándote cuenta de que, para realizar una modificación o añadido, muchas veces necesitas desarrollar muchas más funcionalidades, lo que conlleva volver a realizar el análisis y diseño de la totalidad del sistema para llevar a cabo esa nueva idea.

Para este tipo de proyectos es donde realmente he visto útil el haber cursado asignaturas de ingeniería de software que en su día se hacían tan pesadas al realizar tanta documentación redactando requisitos y realizando diagramas.

6.5. TRABAJOS FUTUROS

Partiendo de las aplicaciones desarrolladas, se podrían realizar trabajos posteriores como por ejemplo que la aplicación móvil automáticamente realice modificaciones en los parámetros definidos de envío y captura para hacer un uso más eficiente de los recursos del dispositivo móvil; en caso de que detectara que el dispositivo se encuentra mucho tiempo en reposo, que sea la propia aplicación la que aumente en “x” el intervalo de captura de los datos de los sensores; si en el nuevo tiempo de captura definido los valores que se registran indican que se encuentra aún en reposo, aumentar de nuevo en “x” el intervalo de captura actual y de ese modo hasta que se detectara un cambio en el patrón de los sensores volviendo de nuevo al intervalo de captura inicial.

Con motivo del aumento del intervalo de captura, no tendría sentido que el intervalo de envío fuera el mismo, por lo que habría que aumentarlo también en “y” por cada uno de los aumentos de tiempo que sufre el intervalo de captura para luego volver al intervalo de envío inicial.

Gracias a que la comunicación entre la aplicación móvil y web se realizan por medio de *Google Cloud Endpoint*, otra ampliación sería desarrollar la aplicación móvil para dispositivos móviles con sistema operativo iOS; únicamente haría falta adecuar la aplicación al lenguaje de programación de dicha plataforma, e incluir las librerías generadas de los *endpoints* para la comunicación con el servidor, ya que sería totalmente compatible el código actual del servidor.

Por último, utilizando los datos obtenidos y con la ayuda de algoritmos avanzados, se podría desarrollar un sistema de desbloqueo del terminal realizando algún gesto característico por parte del usuario; por ejemplo, realizando un giro con el dispositivo.

REFERENCIAS

1. BOE. *BOE*. [En línea] [Citado el: 10 de Febrero de 2014.] artículo 3.a. <http://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf>.
2. Android developer. *Android developer*. [En línea] [Citado el: 26 de Enero de 2014.] Guías para desarrollo de aplicaciones Android. developer.android.com/intl/es/guide/topics/sensors/sensors_overview.html.
3. **The MathWorks, Inc.** The MathWorks, Inc. *The MathWorks, Inc.* [En línea] [Citado el: 10 de Febrero de 2014.] <http://www.mathworks.com/matlabcentral/fileexchange/40876-android-sensor-support-from-matlab/content/sensorgroup/Examples/html/CapturingAzimuthRollPitchExample.html>.
4. eHow en Español. *eHow en Español*. [En línea] [Citado el: 06 de Febrero de 2014.] www.ehowenespanol.com/giroscopio-acelerometros-usados-telefonos-inteligentes-info_189829/.
5. Biblioteca de Ingeniería, Universidad de Sevilla. *Biblioteca de Ingeniería, Universidad de Sevilla*. [En línea] [Citado el: 06 de Febrero de 2014.] bibing.us.es/proyectos/abreproy/11638/fichero/Capitulo+4.pdf.
6. **Giménez, Carlos.** [En línea] [Citado el: 06 de Febrero de 2014.] www.jeuazarru.com/docs/Acelerometro.pdf.
7. **Móvil, Sistema de Alarma.** Google Play. *Google Play*. [En línea] 1.3.1. [Citado el: 10 de Febrero de 2014.] <https://play.google.com/store/apps/details?id=com.alarmsystem.focus&hl=es>.
8. **Shake2PlayNext.** Google Play. *Google Play*. [En línea] 1.3.3. [Citado el: 10 de Febrero de 2014.] <https://play.google.com/store/apps/details?id=ru.atrant.shake2playnext&hl=es>.
9. AprendeAndroid. *AprendeAndroid*. [En línea] [Citado el: 06 de Febrero de 2014.] www.aprendeandroid.com/l9/sensores_android_posicion.htm.
10. Todo Interesante. *Todo Interesante*. [En línea] [Citado el: 06 de Febrero de 2014.] www.todointeresante.com/2009/04/que-es-un-giroscopio-como-funciona-y.html.
11. Wikipedia, La enciclopedia libre. *Wikipedia, La enciclopedia libre*. [En línea] [Citado el: 06 de Febrero de 2014.] es.wikipedia.org/wiki/Sistema_de_posicionamiento_global.
12. Moves. *moves*. [En línea] [Citado el: 27 de Enero de 2014.] www.moves-app.com.
13. Robaid. *Robaid*. [En línea] [Citado el: 27 de Enero de 2014.] www.robaid.com/bionics/t-shirt-with-sensors-and-a-smart-system-monitor-health.htm.
14. Apple. *Apple*. [En línea] [Citado el: 05 de Febrero de 2014.] <http://www.apple.com/es/iphone-5s/features/>.
15. Agencia Española de Protección de datos. [En línea] [Citado el: 10 de Febrero de 2014.] http://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Guías/GUÍA_SEGURIDAD_2010.pdf.
16. Wikipedia, La enciclopedia libre. *Wikipedia, La enciclopedia libre*. [En línea] [Citado el: 27 de Enero de 2014.] [es.wikipedia.org/wiki/Java_\(lenguaje_de_programación\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programación)).
17. **Twitter.** Bootstrap. *Bootstrap*. [En línea] [Citado el: 26 de Enero de 2014.] <http://getbootstrap.com/about>.
18. Wikipedia, La enciclopedia libre. *Wikipedia, La enciclopedia libre*. [En línea] [Citado el: 26 de Enero de 2014.] [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)).

19. Google developers. *Google developers*. [En línea] [Citado el: 27 de Enero de 2014.] developers.google.com/appengine/docs/whatisgoogleappengine?hl=es.
20. Android developer. *Android developer*. [En línea] [Citado el: 27 de Enero de 2014.] developers.google.com/appengine/docs/java/endpoints.
21. OpenSignal. *OpenSignal*. [En línea] [Citado el: 27 de Enero de 2014.] Datos de fragmentación en Android. opensignal.com/reports/fragmentation-2013/.
22. IDC, Analyze the Future. *IDC, Analyze the Future*. [En línea] [Citado el: 27 de Enero de 2014.] Cuota sistemas operativos. www.idc.com/getdoc.jsp?containerId=prUS24442013.
23. JavaNB. *JavaNB*. [En línea] [Citado el: 27 de Enero de 2014.] book.javanb.com/Professional-Java-Development-with-the-Spring-Framework/BBL0095.html.
24. Androideity. *Androideity*. [En línea] [Citado el: 27 de Enero de 2014.] androideity.com/2012/05/10/la-importancia-del-mvc-en-android/.
25. **Cebollero, David Díez**. *Diseño de Sistemas Interactivos*. [Apuntes Universidad Carlos III de Madrid]. Marzo de 2013.
26. Wikipedia, La enciclopedia libre. *Wikipedia, La enciclopedia libre*. [En línea] [Citado el: 04 de Febrero de 2014.] es.wikipedia.org/wiki/Diagrama_de_componentes.

ANEXO A – GESTIÓN DE PROYECTO

A continuación se presenta la planificación definida para cumplir los plazos estipulados para la entrega del mismo y por último el presupuesto incluyendo los gastos de personal y material necesarios para su consecución.

Para realizar la planificación se ha utilizado un diagrama de Gantt con cada una de las fases definidas en el desarrollo y las tareas más importantes identificadas en cada una de ellas, así como el coste en días que ha llevado la realización de cada una.

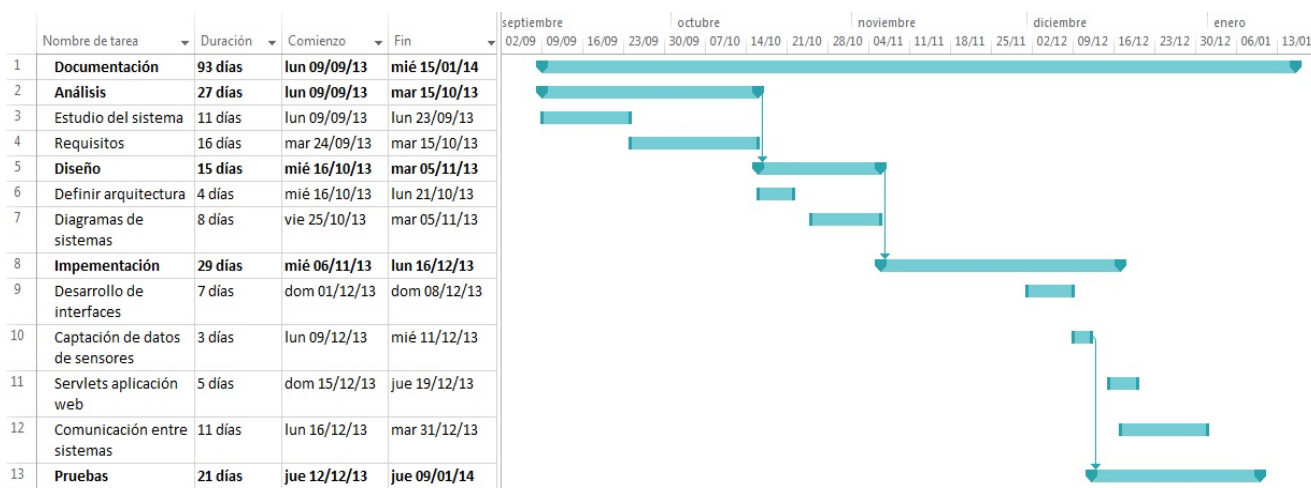


Imagen 41. Diagrama de Gantt - Planificación.

Como se observa en la imagen anterior, el proyecto comenzó el día 9 de Septiembre de 2013 con el propósito de acabarlo la primera quincena de Enero de 2014. En concreto, en la planificación se indicó que se terminaría el 15 de Enero de 2014. La fase de documentación se alarga durante todas las fases del proyecto ya que continuamente se alterna cada una de las mismas con la documentación que le corresponde, y el proyecto concluye cuando finalmente se termina de documentar todo el trabajo realizado. El total de días planificados son 93, aproximadamente tres meses.

Para realizar el análisis económico, se utiliza el tiempo en días obtenido en la planificación del proyecto para calcular la amortización de los materiales utilizados, así

como de los gastos de personal a los que habría que hacer frente. Para ello, a continuación se detallan los costes:

Perfil	Salario bruto anual (€)	Coste anual (€)	Coste mes (€)	Coste hora (€)
Ingeniero Informático	30.000,00	39.000,00	3.545,45	22,16

Tabla 59. Perfiles de personal.

Para el cálculo del coste real anual, hay que añadir el coste de la Seguridad Social; en este caso se considera un incremento de un 30% del salario bruto anual. Posteriormente, se divide el coste anual entre los 11 meses productivos del año. El coste hora se obtiene de establecer que se dedica al proyecto 160 horas al mes.

Duración	Meses aprox.	Horas/mes	Coste Total (€)
93 días	3	160	10.636,80

Tabla 60. Coste de personal - Planificado.

El coste total se obtiene a partir de la multiplicación de los meses aproximados por las horas al mes trabajadas por su coste hora.

$$\text{Coste Total} = ((3 * 160) * \text{CosteHora})$$

Al coste del personal hay que añadirle el coste de todos los equipos necesarios para el desarrollo del proyecto, así como los programas y costes generales a tener en cuenta, que en este caso serán: la luz, el agua y el internet. Todos los importes especificados a continuación no incluyen I.V.A.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

Equipo	Cantidad	Precio/unidad	Vida útil (meses)	Tiempo de uso (meses)	Coste (€)
iMac 21'5" 2'5GHz Intel Core i5	1	973,73	48	3	60,85
HP Pavilion dv-6	1	800,00	48	3	50,00
Samsung Galaxy S3	1	378,41	48	3	23,65
TOTAL					134,50

Tabla 61. Coste de equipos - Planificado.

Software	Cantidad	Precio(€) /unidad	Vida útil (meses)	Tiempo de uso (meses)	Coste (€)
Windows 7 Professional	1	103,31	48	3	6,45
Project Professional 2013	1	1081,51	48	3	67,59
Google App Engine	1	*	-	3	3,00
POV-Ray v3.7	1	0,00	-	1	0,00
Visual Paradigm for UML 10.1	1	0,00	-	1	0,00
Paint	1	0,00	-	2	0,00
Microsoft Office	1	539,00	48	3	33,68
TOTAL					110,72

Tabla 62. Coste de software - Planificado.

*El coste está asociado al uso que se haga de él.

El coste aplicado tanto en los programas software como en los equipos es el correspondiente a la duración del proyecto, teniendo en cuenta la vida útil estimada para cada uno de ellos.

Los costes generales asociados al proyecto sin I.V.A son los siguientes:

Concepto	Coste (€)/mes	Meses	Coste (€)
Agua	20,00	3	60,00
Luz	40,00	3	120,00
Internet	25,00	3	75,00
TOTAL			255,00

Tabla 63. Costes generales - Planificado.

Los costes totales del proyecto es la suma de todas las cantidades anteriores aplicando un margen de error en la estimación de un 10% sobre el importe acumulado. Además de ese margen de error, hay que aplicar un incremento de un 25% para obtener beneficios y por último añadirle el I.V.A aplicable que es de un 21% según la legislación actual. Dicha cantidad queda reflejada en la siguiente tabla:

Concepto	Coste (€)
Personal	10.636,80
Hardware	134,50
Software	110,72
Generales	255,00
Margen de error (10%)	1.113,70
Margen de beneficio (25%)	3.062,68
Total sin I.V.A	15.313,40
I.V.A (21%)	3.215,81
Total	18.529,21

Tabla 64. Resumen costes - Planificado.

El importe final del proyecto asciende a **dieciocho mil quinientos veintinueve euros con veintiún céntimos.**

PLANIFICACIÓN REAL

La planificación establecida al comienzo del proyecto ha sufrido pequeñas desviaciones, lo que ha provocado que la diferencia sea de haberlo concluido con 20 días de retraso respecto a la primera idea.

Como se puede observar en la siguiente imagen, hasta la fase de análisis incluida, se han cumplido con éxito los plazos, pero es a partir de esa fase cuando comienzan las desviaciones en la planificación.

En la inicial se determinó que el número de horas dedicadas al mes serían 160, pero finalmente se han dedicado menos horas por diversos motivos que al final del anexo se analizarán.

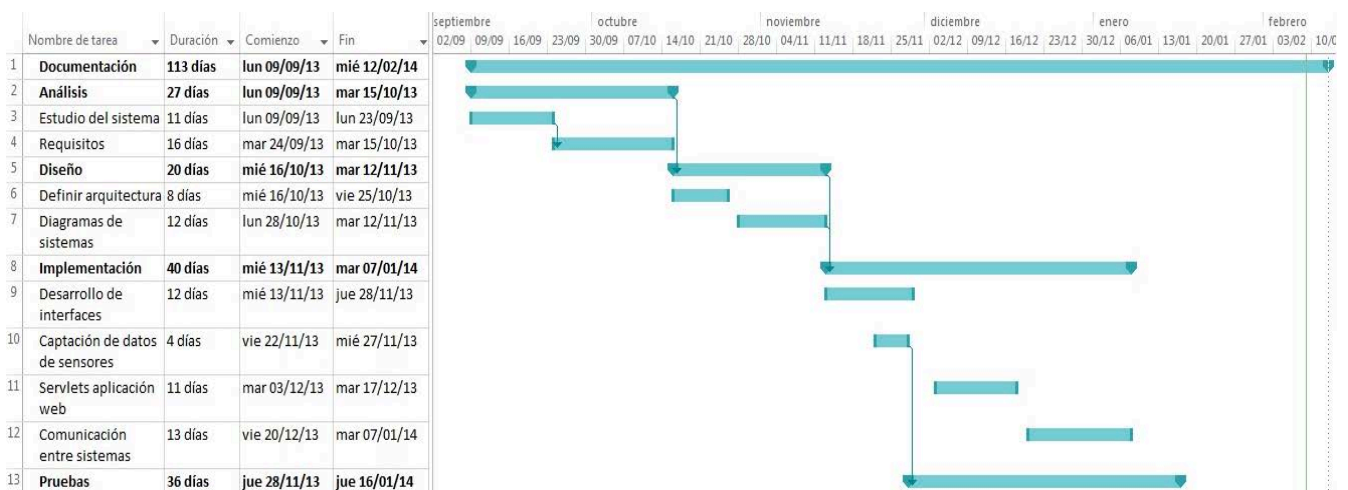


Imagen 42. Diagrama de Gantt - Real.

Duración	Meses aprox.	Horas/mes	Coste Total (€)
113 días	4	100	8.864,00

Tabla 65. Coste de personal - Real.

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

Equipo	Cantidad	Precio/unidad	Vida útil (meses)	Tiempo de uso (meses)	Coste (€)
iMac 21'5" 2'5GHz Intel Core i5	1	973,73	48	4	81,14
HP Pavilion dv-6	1	800,00	48	4	66,66
Google Nexus 5 32GB	1	315,21	48	2	13,13
Samsung Galaxy S3	1	378,41	48	2	15,76
TOTAL					176,69

Tabla 66. Coste de equipos - Real.

Software	Cantidad	Precio(€) /unidad	Vida útil (meses)	Tiempo de uso (meses)	Coste (€)
Windows 7 Professional	1	103,31	48	4	8,60
Project Professional 2013	1	1081,51	48	4	90,12
Google App Engine	1	*	-	4	3,00
POV-Ray v3.7	1	0,00	-	1	0,00
Visual Paradigm for UML 10.1	1	0,00	-	1	0,00
Paint	1	0,00	-	1	0,00
Microsoft Office	1	539,00	48	4	44,91
TOTAL					146,63

Tabla 67. Coste de software - Real.

. *El coste está asociado al uso que se haga de él.

Concepto	Coste (€)/mes	Meses	Coste (€)
Agua	20,00	4	80,00
Luz	40,00	4	160,00
Internet	25,00	4	75,00
TOTAL			315,00

Tabla 68. Costes generales - Real

Con respecto a la planificación inicial, todo el material y costes generales se han incrementado debido a que se ha retrasado casi un mes la finalización del proyecto. Esto se debe a que, por motivos de salud y laborales, no se han podido cumplir los plazos establecidos al comienzo del proyecto, además de incrementarse el número de dispositivos hardware para la realización de pruebas. Puede parecer que con esto que no se consiguen beneficios, pero no es así, principalmente porque el mayor coste que tiene es el de personal, y las horas dedicadas por mes han sido inferiores reduciéndose éste de los 10.636,80€ a 8.864,00€ reales. A continuación se muestra la diferencia de costes planificados con los iniciales. Todas las cantidades son sin I.V.A.

Concepto	Planificado (€)	Real (€)
Personal	10.636,80	8.864,00
Hardware	134,50	176,69
Software	110,72	146,63
Generales	255,00	315,00
Total	11.137,02	9.502,32

Tabla 69. Diferencia de costes.

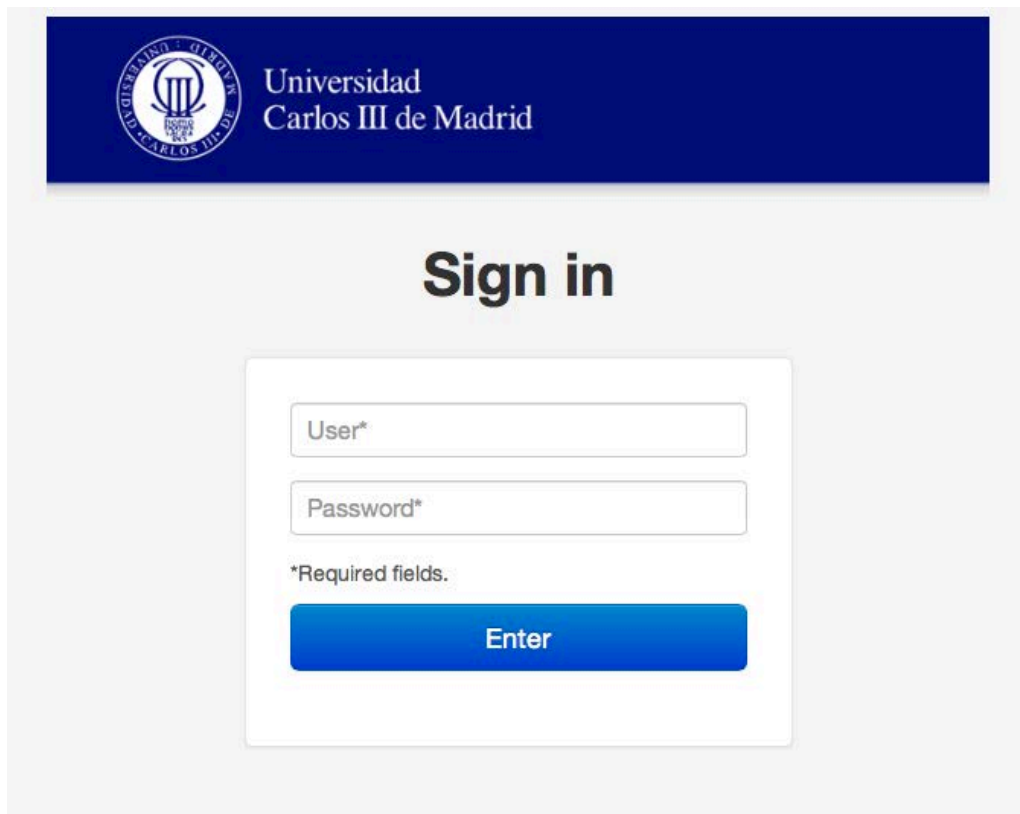
Como se observa en la tabla anterior, la diferencia es nada más de 1.634,70€ de diferencia respecto a los costes planificados, por lo que, no solo se consiguen beneficios sino que se ha conseguido un ahorro de los costes aumentando el beneficio.

ANEXO B – MANUAL DE USUARIO

En este anexo se mostrarán las pantallas que componen tanto la aplicación web como la móvil y se explicará cómo interactuar con cada una de ellas para conocer todas las posibilidades que ofrecen.

A.1 APLICACIÓN WEB

La aplicación web se compone de tres pantallas principales: la de acceso, la de consulta de datos almacenados en el datastore y la pantalla para modificar los parámetros de captura y envío de datos de los sensores al servidor.



Universidad Carlos III de Madrid

Sign in

User*

Password*

*Required fields.

Enter

Imagen 43. Pantalla de acceso a la aplicación web.

A través de esa primera pantalla, el administrador deberá introducir un nombre de usuario y contraseña correctos y a continuación pulsar “Enter”. En el caso de que los datos introducidos sean correctos, se accederá a la pantalla de administración.



Imagen 44. Pantalla administración.

En el caso de que sean incorrectos, se mostrará un error.

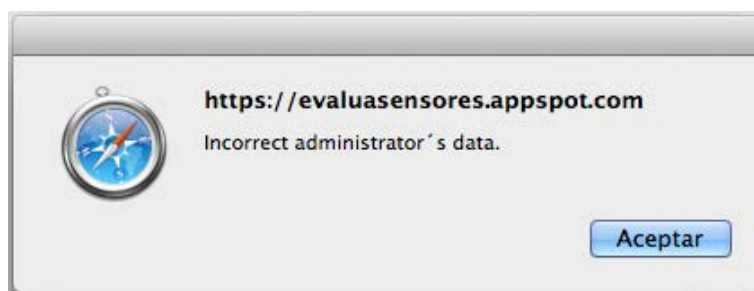


Imagen 45. Acceso incorrecto.

UNIVERSIDAD CARLOS III

APLICACIÓN MÓVIL PARA LA CAPTURA DESATENDIDA DE DATOS DE SENSORES EN
TELÉFONOS INTELIGENTES

Una vez se encuentre en la pantalla de administración, se pueden llevar a cabo dos acciones, consultar datos y ajustes en los clientes que tienen instalada la aplicación móvil y se encuentran dados de alta en el servidor.

Universidad Carlos III de Madrid

Welcome pepin !! Last connection: 10 Feb 2014 16:57:28 [Log out](#)

CONTENT

- Home
- Data**
- Settings

Data search

User ID

Data shown:

☐ GPS

☐ Gyroscope

☐ Position

☐ Accelerometer

Results

MessageID	UserID	Mode	CaptureIv	Timestamp
-----------	--------	------	-----------	-----------

Imagen 46. Pantalla para consultar datos.

En la pantalla de consulta de datos, cuando se realiza la búsqueda por el id de un usuario que se encuentre en la base de datos, se muestran los resultados en la zona “Results”.

Results				
MessageID	UserID	Mode	CaptureIv	Timestamp
Mensaje(5258163104251904)	5801253128896512	GENERAL	100	Sat Jan 11 20:53:13 CET 2014 20 53 13
Mensaje(5295598039203840)	5801253128896512	GENERAL	100	Sat Jan 11 20:03:22 CET 2014 20 3 22
Mensaje(5304651830263808)	5801253128896512	PASEO	100	Sat Jan 11 20:53:22 CET 2014 20 53 22
Mensaje(5840251431944192)	5801253128896512	GENERAL	100	Sat Jan 11 20:39:49 CET 2014 20 39 49
Mensaje(5846255796224000)	5801253128896512	GENERAL	100	Sun Jan 12 10:57:42 CET 2014 10 57 42

Imagen 47. Resultados consulta.

La última pantalla contiene el formulario para modificar los parámetros de los clientes, y se accede a ella a través de la opción “Settings” de la barra de contenidos.

Universidad Carlos III de Madrid

Welcome pepin !! Last connection: 11 Feb 2014 17:30:52 [Log out](#)

CONTENT

- Home
- Data
- Settings**

Configuration

Capture interval (ms)

Send Interval (hours)

Confirm

Imagen 48. Pantalla modificación de clientes.

Una vez introducidos los datos, únicamente hay que pulsar el botón “Confirm”.

A.2 APLICACIÓN MÓVIL

La aplicación desarrollada se caracteriza principalmente por su facilidad de uso, estando formada únicamente por una pantalla y la perteneciente al registro. A continuación se muestra la pantalla principal y las posibles acciones que se pueden llevar a cabo en ella.

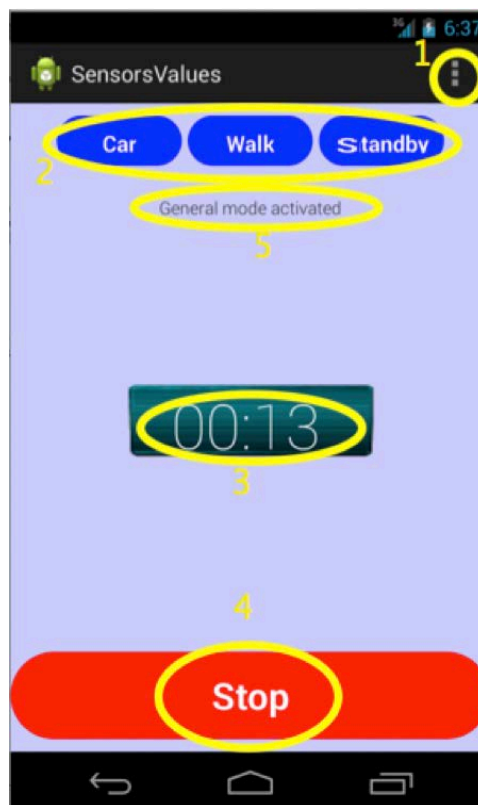


Imagen 49. Pantalla principal.

- 1.- Pinchando en el menú de ajustes da la posibilidad de darse de baja.
- 2.- Botones de cambio de modo.
- 3.- Muestra el tiempo de ejecución de la aplicación en un modo diferente a "inactive".
- 4.- Permite parar la lectura de los sensores entrando en modo "inactive".
- 5.- Información del modo actual de ejecución.

Si el usuario entra en el menú de ajustes de la aplicación y elige la opción “Log out”, a continuación se le mostrarán los avisos de las imágenes siguientes:

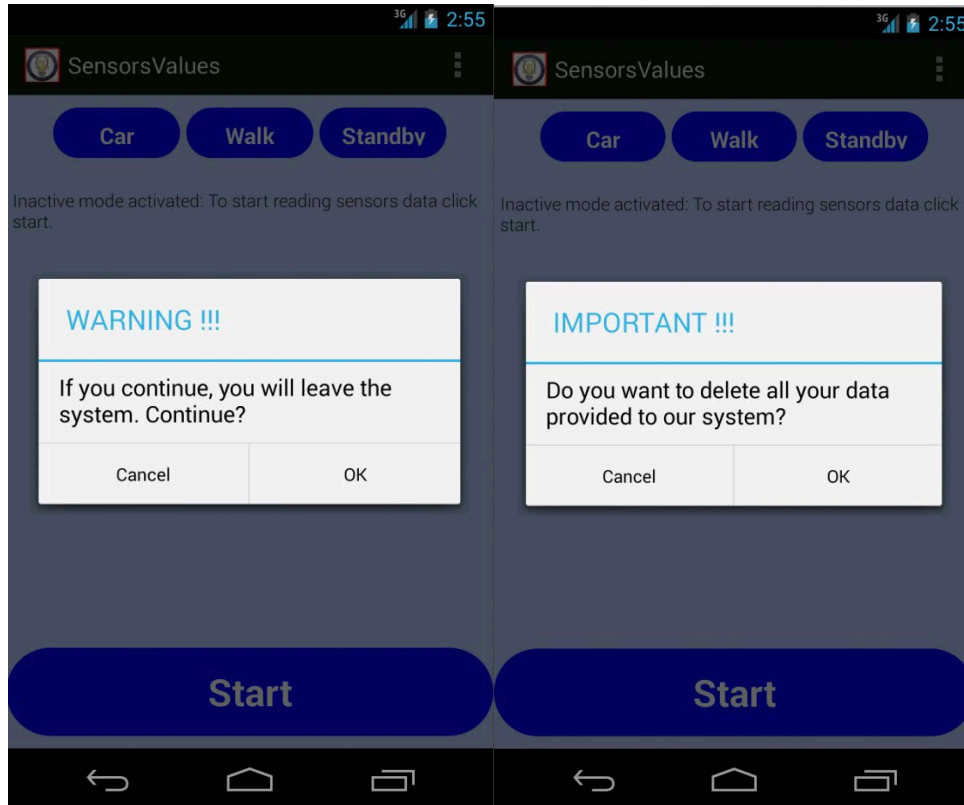


Imagen 50. Baja de aplicación móvil.

En la imagen de la izquierda, se avisa para que el usuario confirme la acción por si no está seguro. Si se cancela, se vuelve a la pantalla principal, mientras que si se pulsa “OK”, se mostrará la imagen de la derecha preguntando si se quiere borrar los datos de los sensores recogidos por la aplicación.

A continuación se muestra la pantalla una vez el usuario se ha dado de baja en el sistema. En el caso de que quiera volver a darse de alta en el sistema, deberá pulsar el botón “Sign up”.



Imagen 51. Pantalla de registro.